Three-Dimensional Static and Dynamic Reactor Calculations by the Nodal Expansion Method

Christensen, Brian

Publication date:
1985

Document Version
Publisher's PDF, also known as Version of record

Risø-R-496

RISØ

Risø-R-496

Three-dimensional Static and Dynamic Reactor Calculations by the Nodal Expansion Method

Brian Christensen

Risø National Laboratory, DK-4000 Roskilde, Denmark
May 1985

RISØ-R-496

THREE-DIMENSIONAL STATIC AND DYNAMIC REACTOR CALCULATIONS BY THE NODAL EXPANSION METHOD

Brian Christensen

This report is written for partial fulfilment of a Ph.D. degree.

Abstract. This report reviews various method for the calculation of the neutron-flux- and power distribution in an nuclear reactor. The nodal expansion method (NEM) is especially described in much detail. The nodal expansion method solves the diffusion equation. In this method the reactor core is divided into nodes, typically 10 to 20 cm in each direction, and the average flux in each node is calculated. To obtain the coupling between the nodes the local flux inside each node is expressed by use of a polynomial expansion. The expansion is one-dimensional, so inside each node such three expansions occur. To calculate the expansion coefficients it is necessary that the polynomial expansion is a solution to the one-dimensional diffusion equation. When the one-dimensional diffusion equation is established a term with the transversal leakage occur, and this term is expanded after the same polynomials. The resulting equation system with the expansion coefficients as the unknowns is solved with weighted residual technique.

(Continued next page)

May 1985
Risø National Laboratory, DK-4000 Roskilde, Denmark

The nodal expansion method is built into a computer program (also called NEM), which is divided into two parts, one part for steady-state calculations and one part for dynamic calculations. It is possible to take advantage of symmetry properties of the reactor core. The program is very flexible with regard to the number of energy groups, the node size, the flux expansion order and the transverse leakage expansion order. The boundary of the core is described by albedos. The program and input to it are described.

The program is tested on a number of examples extending from small theoretical ones up to realistic reactor cores. Many calculations are done on the wellknown IAEA benchmark case. The calculations have tested the accuracy and the computing time for various node sizes and polynomial expansions. In the dynamic examples various strategies for variation of the time step-length have been tested.

INIS-Descriptors: COMPUTERIZED SIMULATION; N CODES; NEUTRON FLUX; POLYNOMIALS; POWER DISTRIBUTION; REACTOR CORES; REACTOR KINETICS; SERIES EXPANSION; THREE-DIMENSIONAL CALCULATIONS

ISBN 87-550-1169-1
ISSN 0106-2840

Grafisk Service Center, Risø 1985

CONTENTS

Page

NOMENCLATURE

Symbol	Dimension	Meaning
a	-	$\frac{D}{\Sigma_u}$
A	-	final coefficient, A-coefficients
A	-	coefficient matrix
A_x^m	cm^2	surface area of node m perpendicular on x
B	-	coefficient matrix
B	cm^{-1}	right side in the equation system for determination of the higher-order coefficients
B_z	cm^{-1}	the buckling
c	-	expansion coefficients for the one-dimensional flux
C_i	cm^{-3}	concentration of delayed neutrons in family i
d	cm	extrapolation distance
d	-	differentiation
D_g	cm	diffusion coefficient in group g
D	-	coefficient matrix
E	-	coefficient matrix
F	-	coefficient matrix
F	-	normalization factor
g	-	energy group
G	-	number of energy groups
G	-	coefficient matrix
I	-	number of delayed neutron families
j	$cm^{-2}\ s^{-1}$	currents
J	$cm^{-2}\ s^{-1}$	current matrix
K	-	coefficient matrix in the equation system for determination of the higher-order coefficients
L	cm^{-1}	transverse leakage
N	-	expansion order

P	-	expansion polynomial
$\mathbf{P}$	-	coupling matrix
$\bar{r}$	cm	point vector
t	s	time
u	cm	arbitrary coordinate
v	cm	arbitrary coordinate
v	cm s^{-1}	neutron velocity
V	cm^3	node volume
w	cm	arbitrary coordinate
x	cm	x-coordinate
y	cm	y-coordinate
z	cm	z-coordinate
α	-	albedo
β_i	-	fraction of delayed neutrons in family i
γ	-	relaxation parameter
γ	-	elements in gamma matrix
Γ	-	gamma matrix
Δ	-	delta, difference, differential operator
λ	-	eigenvalue
λ_i	cm^{-1}	decay constant
Λ	-	normalization constant
ν	-	average number of neutrons produced in a fission
ξ	cm^{-3}	one-dimensional distribution of delayed neutrons
Σ	cm^{-1}	cross section
$\phi_g(\bar{r},t)$	cm^{-2} s^{-1}	flux in group g at point $\bar{r}$ to time t
$\phi_g^m(t)$	cm^{-2} s^{-1}	flux in group g in node m to time t
χ	-	emission spectrum
ϕ	cm^{-2} s^{-1}	average one-dimensional flux
ω	-	coupling coefficients
ω	-	relaxation parameter
∇	-	nabla, Laplacian operator

Indicies

a	absorption (cross section)
d	delayed (fission spectrum)
f	fission (cross section)
g	energy group
i	delayed neutron family
l	left (current)
n	expansion polynomial order
p	prompt (fission spectrum)
r	removal (cross section)
r	right (current)
s	scattering (cross section)
s	arbitrary direction - left or right
t	total (cross section)
u	arbitrary coordinate - x,y,z

Superscripts

i	delayed neutron family
in	incoming
m	node m
r	inner iteration number
s	outer interation number
out	outgoing
-	negative direction
+	positive direction

1. INTRODUCTION

The idea of this project is to calculate the three-dimensional power distribution in a nuclear power reactor when the geometric and material constants of the reactor are known. During the last twenty years scientists the world over have tried to calculate power distributions and multiplication factors, k_{eff}, for nuclear reactors. Many different methods have been developed with unique advantages and drawbacks. Some computer codes calculate only steady-state problems and others are restricted to two-dimensional problems.

There have been great advances not only in the calculational methods, but also in the power of the computers themselves. Problems which took hours of computer time twenty years ago, today can be managed in a few seconds. It also means that problems which were formerly solved by complicated iteration processes now are solved by direct and perhaps much simpler methods.

On the other hand, the computer work has not been reduced. In earlier years the problems were two-dimensional, but today they are four-dimensional (three-dimensional dynamic problems) which increases the calculational work by several orders of magnitude. In Chapter 2 I have tried to summarize and compare some of the methods which are described in the literature.

The great effort to calculate the power distribution and other properties of a reactor are made because it is very important to be able to predict the behaviour of a real reactor before it is placed into operation. In the design phase the calculation is especially important in order to understand the subcritical and supercritical behaviour of the reactor.

Under normal operating conditions the most economical control rod adjustments and fuel distribution can be simulated and the

reactor made to conform to the model. Burnup can be simulated and an optimum fuel management strategy proposed. Furthermore, with a good overall power distribution calculation, it is essential to be able to predict fuel failures.

Another important area is to simulate the reactor under abnormal conditions, e.g. a control rod ejection. In such cases very severe power transients occur which could have serious effects on the reactor such as local high-temperature rises. The consequences of such an accident have to be simulated to check whether or not the safety system would be able to handle the situation.

A computer code for simulation of a reactor core consists of two nearly distinct parts, which interfere with each other, but in the programming phase they can be considered independent. The first one is the neutronic part, which calculates the neutron flux distribution, the power generation, criticality factors, etc., and it is this part of the code this report treats. The second part is the hydraulic one, which treats the heat transport out of the core and calculates the temperature distribution. This part will not be treated here, except for a single example with feedback from the hydraulics.

Today the Energy Technology Department at Risø National Laboratory bases its overall calculations mainly on the two codes ANTI (Larsen, 1980), (Nielsen and Larsen, 1980) and NOTAM (Schougaard, 1979). The ANTI-program treats PWR's in both steady state and dynamic, and NOTAM is able to treat BWR's steady state. Both codes have their neutronic parts based on the methods used in ANDYCAP (Babala, Bech and Haugset, 1971) and their results are not quite as accurate as desired. The accuracy of their results is rather dependent on the actual problem and also on some empirical constants, which often have to be estimated from other simpler calculations.

Therefore, this work has been initiated with the object of comparing different methods from the literature and finally implementing one at Risø National Laboratory, which is more ac-

curate and preferably quicker than the existing codes. In Chapter 2 I have tried to summarize existing methods and in Chapter 3 the theory of what is called the nodal expansion method is outlined. Chapter 4 describes some of the finer points in the programming, and in Chapter 5 the program has been tested on various benchmarks.

2. METHODS FOR THREE-DIMENSIONAL FLUX CALCULATIONS

Ever since the appearance of the first computers in the fifties, calculations on nuclear reactors have been performed. Much progress both in computers and their methodology has been made in the last thirty years and the codes are now very quick and also very accurate in most cases. One could argue that the development of computers with larger and larger memories, such as CRAY-1 and Cyber-205, could render superfluous further progress in the codes, but this is wrong. At first, the larger computers merely encourage scientists to calculate larger problems; however, power companies and the local power stations perhaps have access only to smaller computers or even mini-computers.

In this chapter some of the methods for flux calculations are described very briefly and an attempt is made to summarize the advantages and disadvantages of the methods. It can be very difficult to compare the different methods, but some of the questions that could be taken into consideration are:

- How effective is the method?
- How flexible is the method?
- Could an error analysis on the result be made?
- Is a good theoretical basis for the method available?
- Does the solution converge to the exact solution of the continuous problem for mesh refinement?

- Is the method reliable?
- Is the method user friendly with respect to inclusion of feedback effects?

Not all these questions can be answered for the methods, e.g. It is nearly always impossible or in any case very difficult to perform an error analysis. It is also difficult to compare the accuracy and speed of various methods, because different computers are used, where calculating times cannot be compared directly. Besides, one method may be able to handle one problem very well (it may be optimized to that problem), while it is hardly able to handle another.

The method that is described below is mainly based on diffusion theory and the diffusion equation:

$$\nabla \cdot D\nabla\phi - \Sigma_t \cdot \phi + S = \frac{1}{v} \cdot \frac{d\phi}{dt} \qquad (2.1)$$

For very accurate calculations Boltzmann's transport equation is a better starting point, but until now and with the available computers, it is too large a task for three-dimensional overall calculations.

The different methods can be divided roughly into five categories:

1. Finite-difference methods (FDM)
2. Flux-synthesis methods (FSM)
3. Finite-element methods (FEM)
4. Response-matrix methods (RMM)
5. Nodal methods (NM)

where the main difference is the independent variables. Finite-difference methods and flux-synthesis methods use point fluxes, while finite-element methods use expansion coefficients. In the response-matrix methods the primary variables are the partial currents, and in the nodal methods they are the average node fluxes. Some methods can combine more of the above-mentioned methods and it can be difficult to classify them.

2.1. Finite-difference Methods

One of the first used and also one of the simplest methods for solving the diffusion equation is a finite-difference approximation for the differential operator. The reactor core is divided into a number of meshes, and the differential operator is approximated with a five-point formula in the two-dimensional case and a seven-point one for three dimensions. It results in an equation system of the form

$$\sum_j \omega_{ij}(\phi_j-\phi_i) - a_i\phi_i + S_i = \frac{d\phi_i}{dt}$$

where

- i is the index of the center mesh and
- j are indexes of the adjacent meshes
- ω_{ij} are the coupling constants between the meshes and are rational functions of the diffusion coefficents D and the mesh lengths.

The resulting equation system can be solved with well-known iteration techniques including different acceleration methods such as successive overrelaxation, coarse-mesh rebalancing and others. The finite-difference methods are used in the codes PDQ-7 (Cadwell, 1967), VENTURE (Vondy, 1977) and in our own TWODIM (Lindstrøm Jensen, 1971).

The advantages of the finite-difference method are that it is very flexible with respect to reactor types, geometry, symmetry properties, etc. and that the theory for the method is well established. The solutions converge to the exact solution for the continuous problem for smaller and smaller mesh, but the main disadvantage is also that small meshes are required to attain an acceptable solution. Werner (1975) reported that acceptable results for LWR's required meshes less than 3-4 cm and Siewers' and Jager's (1977) results with the code XYZ-MUGDI

showed that the mesh-size should be less than 1.5 cm to produce reliable results.

With such small mesh sizes the FDM is practically useless for large three-dimensional calculations, because of the enormous number of unknowns. On the other hand, the method is good as reference code, because of the well-founded theory and also because the local flux-distribution is directly attained.

2.1.1. The Coarse Mesh Finite-difference Method

The finite-difference method can be improved by using a higher-order approximation for the differential operator, e.g. a nine-point formula in the two-dimensional case. This gives a coupling to both the neighbour meshes and the next neighbour meshes. In theory this should be a better approximation, but for practical calculations there are no great advantages with a higher-order formula.

Another attempt to improve the finite-difference method has been made by Børresen (1981). He uses the simplest FDM together with two additional approximations

$$\frac{D_i \cdot D_j}{D_i + D_j} \approx \frac{1}{2} \sqrt{D_i} \cdot \sqrt{D_j}$$

and

$$\bar{\phi} = a \cdot \bar{\phi}_i + \frac{1-a}{6} \sum_j \phi_{ij}$$

where

a is a constant

$\bar{\phi}_i$ is the node average flux

ϕ_{ij} is the flux on the interface between node i and j

With these two approximations the number of variables to be stored is much smaller than for a conventional FDM and with the tunable parameter a, the method can use meshes of assembly-size or half assembly-size.

Siewers and Jager (1977) reported that their code ROSI, which is based on the same theory, produces accurate results for mesh sizes in the range of 3 to 10 cm. Børresen uses the method in the code PRESTO and it is reported to be extremely fast and rather accurate.

One of the disadvantages of the method is that it uses an empirically determined constant a, which has to be calculated by other methods, when a new problem shall be solved. Secondly, the theoretical background for the method is poor.

2.1.2. The Flux-expansion Method

A higher-order coarse mesh FDM, which is described by Langenbuch et. al. (1977a), uses a method much like finite-element techniques. The reactor is assumed to be composed of rectangular mesh boxes and the fluxes inside each box are approximated by polynomials in the x, y, and z direction. The polynomials

$$G_x = a_x \cdot \xi_x + b_x \cdot \xi_x^2 + c_x \cdot \xi_x^3 + \ldots.$$

where the independent variables are expressed in dimensionless mesh-centered variables, defined in terms of the mesh spacings h

$$\xi_x = \frac{x - x_i}{h_{xi}}$$

are used in the flux approximations.

The fluxes inside each mesh are approximated in one of two ways, either the separable Sum-method

$$\phi(x,y,z) = \phi_{ijk} \cdot (1+G_x+G_y+G_z)$$

or the inseparable Product-method.

$$\phi(x,y,z) = \phi_{ijk} \cdot (1+G_x) \cdot (1+G_y) \cdot (1+G_z)$$

The polynomial expansion can be stopped with second order, which gives the QUAD methods, or after third order, which gives the CUBE methods.

The flux expansion is inserted in the diffusion equation and by demanding continuity in the flux expansion with the adjacent meshes some of the expansion coefficients can be determined.

The rest of the coefficients are determined by weighted residual techniques in the codes QUABOX and CUBBOX (Langenbuch, Maurer and Werner, 1977a) or by least-squares fit in the code NOXE3D (Rydin and Sullivan, 1978).

The final equation system is very much like the system for the simple FDM, and the same acceleration techniques can be used, but the method can use much larger meshes (10-20 cm). Langenbuch et al. (1977 b) reported very accurate results for calculations with large meshes on the IAEA-2D benchmark (Micheelsen and Neltrup, 1973), but they also use high expansion order (6), and here is the only drawback of the method: The theoretical basis for the use of high-order expansions is relatively poor.

2.2. Flux-synthesis Methods

Another way of dealing with the problem of the enormous number of unknowns is to use the physical fact that a reactor is more homogeneous in the vertical than in the horizontal direction. In the flux-synthesis methods the spatial flux is assumed separable

$$\phi(x,y,z) = \phi(x,y) \cdot \phi(z)$$

which reduces the three-dimensional problem to two- and one-dimensional ones.

A more general class of the synthesis-methods are based on the approximation

$$\phi(x,y,z) = \sum_{n=1}^{N} a_n(z) \cdot \psi_n(x,y)$$

where

$a_n(z)$ are the mix functions and

$\psi_n(x,y)$ are the radial flux solutions also called the trial-functions.

The trial-functions have to be found by other methods, e.g. by finite-difference methods. The mix functions can then be found by various methods, e.g. moment weighting (Galerkin) methods, or with the use of variational principles (Selengut).

At Risø, we have a code called SYNTRON (Larsen, 1971), which uses Selengut weighting in the solution process. The flux-synthesis method is very fast and does not need much computer storage. The disadvantage of the methods lies in the assumption of separable flux in the horizontal and vertical directions. This assumption is especially unfulfilled for light water reactors, where the presence of control rods causes severe flux tilts.

Another approach to the flux synthesis-method called the multi-channel fluxsynthesis method was made by Wachspress et al. (1972). In this method the spatial flux approximation can be written as

$$\phi(x,y,z) = \sum_{n=1}^{N} \sum_{c=1}^{C} a_{nc}(z) \cdot f_{nc}(x,y) \cdot \psi_n(x,y)$$

which differs from the preceding one by the additional sums over channels (c for channels) and the set of multichannel basis-functions $f_{nc}(x,y)$, which permit some modulation of the original expansion functions. This approximation is able to simulate the reactor in more detail, but it also requires a user who knows a great deal about the problem if reliable solutions should be obtained. The user has to find good trial functions as well as choose appropriate channels for the calculation.

2.3. Finite-element Methods

The finite-element technique is well known from other fields of engineering, particularly structural mechanics, and the methods are now also applied to reactor physics. The basis idea is to divide the reactor into meshes called elements, then expand the flux inside each mesh in terms of piecewise polynomials, and consider the expansion coefficients as the primary variables. Continuity conditions along the interfacing of neighbouring elements give the equations for the unknown coefficients.

With the use of piecewise polynomials, also called basisfunctions, it is possible to obtain good approximations of the local flux variation. Two main types of basisfunctions have been proposed for neutron diffusion problems, either Lagrange polynomials, which interpolate between function values at base points distributed evenly over the finite element, or Hermite interpolation polynomials, which are based on function values and derivatives on the boundaries of the element.

Besides, it is possible to model the core boundaries in detail as well as places where great flux changes occur, because the elements do not have to be equilaterals (in two dimensions), but could be triangles or other suitable geometries. The finite-element method is implemented in a code called FEM3D (Franke, 1977), and Franke reported some very good results for the three-dimensional IAEA benchmark.

Recently Jagannathan (1983) reported results obtained with a code FINERC, which is based on a combination of the FEM and the FSM, and this code is a little slower than FEM3D, but much faster than the Risø-code FEM3D (Misfeldt, 1975). Still, the finite-element method is not quite as fast as the familiar coarse mesh finite-difference method (QUABOX-CUBBOX), and some of the nodal methods described later.

The great advantage of the FEM, and the point where it is different from most other methods, is its large flexibility with respect to complicated geometrical configurations and accuracy at critical points. The method works well for relatively large homogeneous regions, but the number of elements or the number of basisfunctions must be increased if heterogeneous problems should be accurately solved. More basisfunctions increase the sizes of the equation systems and also increase the coupling inside each system, which makes them more difficult to solve. Finally, three-dimensional economical finite-element calculations have only begun to be made and nearly no work has been done with three-dimensional dynamic calculations.

2.4. The Response-matrix Method

The response-matrix method (RMM) is characterized by precalculation of response functions, which are matrices relating the partial currents through each of the boundary faces of a coarse mesh. The defining relation is

$$J^{out} = R \cdot J^{in} + J_f$$

where

J^{out} is the outgoing current

J^{in} - - incoming -

J_f - - outgoing - due to external sources

R is the response matrix.

The outgoing currents from one mesh or a part of them will be incoming currents of adjacent meshes, which can be expressed

$$J^{in} = R \cdot J^{out}$$

The final solution can then be obtained by solving these two equations simultaneously.

The calculation of the response matrices is a much smaller problem than the overall calculation, so advanced methods, perhaps based on transport theory, can be used.

The RMM is efficient for problems that can be divided into large meshes, and where the coupling between meshes is not too strong. Inside each mesh the structure may be strongly heterogeneous. The method is formally exact, but for efficient application the method relies on certain assumptions for the spatial and angular distributions of the partial currents.

The RMM has been used in the two-dimensional programs CIKADA and LABAN (Weiss, 1977) and Weiss reported fine results for the two-dimensional IAEA benchmark, but until now I have seen no three-dimensional codes that use the RMM. Perhaps it is because of the rather tight coupling between neighbouring fuel elements in light-water in contrast to heavy-water reactors, where the RMM can be used with great success. The method is poor for dynamic problems, because of feedback mechanisms that complicate the calculations.

2.5. Nodal Methods

In the nodal method the reactor is divided into a number of meshes or nodes and the point flux is integrated over each node. The primary variables are the volume-averaged fluxes and the

surface-averaged currents. The method is formally exact and by the use of integral expressions no singularities occur.

The basis for the nodal theory is the steady-state neutron balance equation:

Production - absorption + leakage in the node
- leakage out of the node = 0

and the coupling between the nodes is obtained with the leakage terms. The nodal method traditionally refers to the following approximation for the leakage:

$$\text{leakage} = \sum_{i \neq j} \ell_{ij}(\phi_i - \phi_j)$$

where ℓ_{ij} is the coupling coefficient between the fluxes at nodes i and j. The different nodal methods are distinguished in their way of determining the coupling coefficients. Some methods use precalculated coupling coefficients based on geometrical data and material parameters, while others make additional calculations frequently under the iterations to attain better values.

Some of the first codes were based on rather empirical formulas for the coupling coefficients, which use adjustable parameters to give the best coincidence with other methods, e.g. with the FDM. With constant coupling coefficients the resulting equation system is simple and can be solved by using different acceleration methods.

At Risø we have two codes, ANTI (Larsen, 1980) and NOTAM (Schougaard, 1979), both of which have their neutronic parts based on the model used in ANDYCAP (Babala et al., 1971). In this model four empirical constants or g-factors are used and they have to be chosen with great care, normally by comparing two-dimensional with TWODIM-calculations. The g-factors are specific for a given reactor and a certain nodesize, but they normally have to be changed if the node size is halved, which also means that the solution does not converge against the continuous solution for mesh-refinements.

The simple nodal methods are very fast, but the solutions obtained are not very accurate, unless careful adjustments have been made, which again make the methods less flexible.

2.5.1. The Nodal Collision Probability Method

From the preceding paragraph it is clearly seen that a better theory for the coupling coefficients has to be applied to the nodal method. Finnemann et al. (1977) defines the node interchanges by partial currents; the stationary diffusion equation can then be written

$$\sum_{u=x,y,z} \frac{1}{a_u} \left[(j^-_{gul} + j^+_{gur}) - (j^+_{gul} + j^-_{gur}) \right] + \Sigma_{tg}\Phi_g$$

$$= \sum_{g'=1}^{G} (\Sigma_{gg'} + \frac{\chi_g}{\lambda} \nu\Sigma_{fg'})\Phi_{g'}$$

where

a_u is the node length in the u-direction

Φ_g is the average flux in group g

j^+_{gul} and j^+_{gur} are average partial currents at the left and right sides of the node.

In the nodal collision probability method the calculation of the partial currents is based on transport theory

$$j^{out}_{gs} = V \sum_{g'=1}^{G} (\Sigma_{gg'} + \frac{\chi_g}{\lambda} \nu\Sigma_{fg'})P_{svgg'} \cdot \Phi_{g'} + \sum_{s'=1}^{6} P_{ss'} \cdot j^{in}_{gs'}$$

where

$P_{svgg'}$ and $P_{ss'}$ are, respectively, the escape and transmission probabilities for the contributing neutrons.

The two equations written above together with some boundary conditions (albedo or symmetry) form a consistent equation system, sufficient to determine the complete set of fluxes and currents. The method is exact if the correct collision probabilities are known, and here lies the major problem of the method.

The collision probabilities depend on the spatial distribution of the sources both in- and outside the node in a very complicated way, which tempts one to make a complex approximation, but on the other hand, the approximation should be simple enough for routine recalculations. Finnemann, Bennewitz and Wagner reported different approximations, but they also reported the major drawback of the method, which is that it does not converge for mesh-refinements. For mesh sizes in the range from 8-20 cm their best approximation gives fairly good results.

2.5.2. The Nodal Expansion Method

In the nodal expansion method the coupling between nodes is based on partial currents just as in the NCPM, but the partial currents are determined here by diffusion theory. The partial currents are calculated with the use of Fick's law

$$j^{+}_{gus} = j^{-}_{gus} - D_g \cdot \frac{d\psi_{gu}}{du}\Big|_s$$

and the diffusion theory expression

$$\psi_{gus} = 2 \cdot (j^{+}_{gus} + j^{-}_{gus})$$

ψ_{gu} is the one-dimensional flux in coordinate u.

The one-dimensional flux is expanded after some polynomials and the lower-order expansion coefficients can be determined by boundary and normalization conditions, while the higher-order terms have to be calculated with weighted residual techniques.

The method is similar to the flux-expansion method and the finite-element method in the flux expansions, but with the difference that it is done with the integral fluxes. The final equation system is similar to the finite-element system, but the coupling matrices are more sparse.

The nodal expansion method is used in the computer program IQSBOX (Finnemann, 1975). In the basic variant the transverse leakage in the one-dimensional calculations is approximated with a flat distribution that shows up to be insufficient for accurate calculations. Later (Finnemann et al., 1977) a linear and a parabolic approximation are used and with this improvement Wagner et al. (1977) report some extremely accurate and fast results for the IAEA-benchmark.

Wagner et al. (1981) claimed that the very accurate determination of the power distribution is valid only for a reactor with fresh fuel, where node-wise constant cross sections and diffusion coefficients can be assumed. In depleted light-water reactors this assumption is not fulfilled, but Wagner et al. describe a non-linear model that is also able to deal with this problem.

The advantages of the nodal expansion method are that it is very fast and accurate, and the solution also converges towards the correct one for mesh-refinements. The only disadvantage is that the theoretical basis is rather poor for higher-order approximations, but for all practical light-water reactors the method works quite well.

2.5.3. The Analytical Method

The basis for the analytical method is the same as for the nodal expansion method. The coupling between nodes is also described by partial currents, but where the NEM uses a polynomial expansion for the one-dimensional flux, the one-dimensional diffusion equation now is solved analytically. The analytical solution to the one-dimensional diffusion equation is then used to determine the partial currents out of each node. The only approximation in this method is a parabolic representation of the transverse leakages in the one-dimensional calculations.

The method has been used in a program called QUANDRY (Smith, 1979) and a lot of very fine results are obtained. The drawback of the method is that the analytical solution method is constrained to two energy groups and to the use of constant cross sections and diffusion coefficients inside each node. Besides, it is rather computer core-consuming, and for solving larger problems we encountered difficulties at Risø.

2.6. Conclusion

In the preceding paragraph I have written a summary of most of the methods that have been used until now. A few methods or combinations of the methods described in the preceding are left out such as the Nodal Synthesis-Method used in HEIDIBLOCK (Lieberoth, 1977) and the Nodal Green's Function Method (Lawrence, 1979).

Among the methods described, three seem to be more efficient than the others: They are the coarse-mesh expansion (QUABOX, CUBBOX), nodal expansion (IQSBOX), and analytical methods (QUANDRY). The codes are reported (Dorning, 1979) to be nearly equally fast and accurate with a slight margin in favour of the nodal expansion method.

Therefore, I have decided to implement a program that is based on the nodal expansion method. It also has the best possibilities of modelling heterogeneous nodes or perhaps introducing Koebkes (1978) discontinuity factors. The analytical method is constrained to two energy groups and besides we already have access to the QUANDRY-code. In comparison with the coarse mesh expansion method, NEM has no particular advantages, except that our old codes ANTI and NOTAM, from which the hydraulic part of the program has to be taken, use nodal methods for the neutronics today.

In Section 3 the nodal expansion method will be described in more detail.

3. THEORY FOR THE NODAL EXPANSION METHOD

3.1. Basic theory for NEM

The nodal expansion method is based on the time-dependent group-diffusion equation for the neutron flux $\phi_g(\bar{r},t)$

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\bar{r},t) = \nabla\cdot D_g(\bar{r},t)\nabla\phi_g(\bar{r},t) - [\Sigma_{ag}(\bar{r},t)+\Sigma_{sg}(\bar{r},t)]\phi_g(\bar{r},t)$$

$$+ \sum_{g'=1}^{G} [\Sigma_{gg'}(\bar{r},t) + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma_{fg'}(\bar{r},t)]\phi_{g'}(\bar{r},t)$$

$$+ \sum_{i=1}^{I} \lambda_i \chi_{dg}^{i} C_i(\bar{r},t)$$

$$g=1,2, \ldots, G \qquad (3.1)$$

where the symbols have the following meanings:

v_g	is the neutron velocity in energy group g.
$\phi_g(\bar{r},t)$	is the neutron flux in group g at location $\bar{r}$ to time t.
D_g	is the diffusion coefficient in group g.
Σ_{ag}	is the absorption cross section in group g.
Σ_{sg}	is the total outscattering cross section from group g, including selfscattering

$$\Sigma_{gg}: \Sigma_{sg} = \sum_{g'=1}^{G} \Sigma_{g,g'}$$

In the following Σ_{ag} and Σ_{sg} is added to Σ_{rg}, which is the total removal cross section.

$\Sigma_{gg'}$ is the scattering cross section from group g' to group g.

Σ_{fg} is the fission cross section in group g.

β is the total fraction of delayed neutrons.

χ_{pg} is the emission spectrum of prompt neutrons in group g.

χ_{dg} is the emission spectrum of delayed neutrons in group g.

λ is an eigenvalue which is unity for a physically critical reactor.

ν is the average number of neutrons produced in a fission.

λ_i is the decay constant of the i'th delayed neutron family.

C_i is the concentration of the i'th delayed neutron family.

In other words, the diffusion equation expresses that

change in flux = -leakage - absorption - outscattering + inscattering + prompt production + delayed production.

The concentrations $C_i(\bar{r},t)$ of the delayed neutron emitters satisfy the following balance equations, which also are called the precursor equations:

$$\frac{\partial}{\partial t} C_i(\bar{r},t) = \frac{1}{\lambda} \beta_i \sum_{g'=1}^{G} \nu\Sigma_{fg'}\phi_{g'}(\bar{r},t) - \lambda_i C_i(\bar{r},t) \qquad (3.2)$$

$$i=1,2, \ldots,I$$

where

β_i is the delayed neutron fraction of the i'th delayed neutron family.

$$\beta = \sum_{i=1}^{I} \beta_i$$

I is the number of delayed neutron families.

Associated with these equations are boundary conditions of the general form

$$\phi_g(R,t) + (R+d)\cdot\nabla\phi_g(R,t) = 0$$

where R is the external reactor boundary and d is the extrapolation distance.

This general form is simplified to one of the following three forms:

$$\phi_g(R,t) = 0$$

$$\tilde{j}_g(R,t) = 0$$

$$j_g^{in}(R,t) = \alpha_g \cdot j_g^{out}(R,t)$$

where

$\tilde{j}_g$ is the net current in group g,

j_g^{in} is the incoming current in group g,

j_g^{out} is the outgoing current in group g,

α_g is the albedo in group g.

The third form can be extended to a more general form by allowing up- and down-scattering at the boundary.

$$j_g^{in}(R,t) = \sum_{g'=1}^{G} \alpha_{gg'} j_{g'}^{out}(R,t)$$

where the elements in the albedo matrix are defined by

$$\alpha_{gg'} = \frac{j_g^{in}(R,t)}{j_{g'}^{out}(R,t)}$$

i.e. $\alpha_{gg'}$ is the fraction of outgoing current in group g' that is scattered back in group g.

The current $\bar{j}_g(\bar{r},t)$ is introduced by Fick's law

$$\bar{j}_g(\bar{r},t) = - D_g(\bar{r},t)\cdot\nabla\phi_g(\bar{r},t) \tag{3.3}$$

If we use Fick's law on the diffusion equation we get the so-called P_1-norm of the diffusion equation

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\bar{r},t) + \nabla\cdot\bar{j}_g(\bar{r},t) + \Sigma_{rg}(\bar{r},t)\phi_g(\bar{r},t)$$

$$= \sum_{g'=1}^{G}\left[\Sigma_{gg'}(\bar{r},t) + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma_{fg'}(\bar{r},t)\right]\phi_{g'}(\bar{r},t) \tag{3.4}$$

$$+ \sum_{i=1}^{I}\lambda_i\chi_{dg}^{i}C_i(\bar{r},t)$$

$$g=1,2,\ldots,G$$

Now the reactor is partitioned into a number of regions or nodes M each with the dimensions Δx, Δy, Δz, and inside each node the cross sections and diffusion coefficients are assumed constant in space. Integrating Eq. (3.4) over each node m, $V = \Delta x\cdot\Delta y\cdot\Delta z$ yields the nodal balance equation.

Integration of Eq. (3.4) term-by-term gives:

The change in flux

$$\int_V \frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\bar{r},t)dV = \frac{1}{v_g}\frac{\partial}{\partial t}\int_V \phi_g(\bar{r},t)dV$$

$$= \frac{1}{v_g}\,V\cdot\frac{d}{dt}\,\phi_g^m(t)$$

The leakage term

$$\int_V \nabla\cdot\bar{j}_g(\bar{r},t)dV = \int_A \bar{j}_g da$$

$$= \int_{A_x} j_g\, dydz + \int_{A_y} j_g\, dxdz + \int_{A_z} j_g\, dxdy$$

$$= \Delta y\cdot\Delta z(j^{+m}_{gxr} - j^{-m}_{gxr} + j^{-m}_{gxl} - j^{+m}_{gxl})$$

$$+ \Delta x\cdot\Delta z(j^{+m}_{gyr} - j^{-m}_{gyr} + j^{-m}_{gyl} - j^{+m}_{gyl})$$

$$+ \Delta x\cdot\Delta y(j^{+m}_{gzr} - j^{-m}_{gzr} + j^{-m}_{gzl} - j^{+m}_{gzl})$$

$$= V \sum_{u=x,y,z} \frac{1}{\Delta u}(j^{+m}_{gur} - j^{-m}_{gur} + j^{-m}_{gul} - j^{+m}_{gul})$$

The removal term

$$\int_V \Sigma_{rg}(\bar{r},t)\phi_g(\bar{r},t)dV = V\cdot\Sigma^m_{rg}(t)\cdot\phi^m_g(t)$$

The inscattering and prompt-production term

$$\int_V [\Sigma_{gg'}(\bar{r},t) + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma_{fg'}(\bar{r},t)]\phi_{g'}(\bar{r},t)dV$$

$$= V\,[\Sigma^m_{gg'}(t) + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma^m_{fg'}(t)]\phi^m_{g'}(t)$$

The delayed neutron production

$$\int_V C_i(\bar{r},t)dV = V\cdot C^m_i(t)$$

The new variables now introduced are

$\phi^m_g(t)$ the average flux in node m and group g at time t.

$C^m_i(t)$ the average concentration of the i'th delayed neutron family in node m.

A^m the surface of node m with the normal pointing out of the node.

j^{+m}_{gus} the average partial currents on the right (s=r) or the left (s=l) surface of node m in the positive direction of coordinate u (u=x,y,z).

j^{-m}_{gus} same as j^{+m}_{gus} but in the negative direction.

The final result of this node-wise integration of Eq. (3.4) is

$$\frac{1}{v_g}\frac{d}{dt}\phi^m_g(t) + \sum_{u=x,y,z}\frac{1}{\Delta u}\left[(j^{-m}_{gul} + j^{+m}_{gur}) - (j^{+m}_{gul} + j^{-m}_{gur})\right] + \Sigma^m_{rg}(t)\phi^m_g(t)$$

$$= \sum_{g'=1}^{G}\left[\Sigma^m_{gg'}(t)+(1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma^m_{fg'}(t)\right]\phi^m_{g'}(t) + \sum_{i=1}^{I}\lambda_i\chi^i_{dg}C^m_i(t) \qquad (3.5)$$

$$g=1,2,\ldots,G$$

In the same way the precursor Eq. (3.2) can be integrated

$$\frac{d}{dt}C^m_i(t)+\lambda_i C^m_i(t) = \sum_{g'=1}^{G}\beta_i\nu\Sigma^m_{fg'}(t)\phi^m_{g'}(t) \qquad (3.6)$$

Integration of Fick's law over a surface A_{ur} yields the terms

$$\int_{A_u} j_g(\bar{r},t)\cdot dvdw = \Delta v\Delta w(j^{+m}_{gur} - j^{-m}_{gur})$$

$$\int_{A_u} D_g(\bar{r},t)\cdot\nabla\phi_g(\bar{r},t)dvdw = \Delta v\Delta w D^m_g(t)\left.\frac{\partial\psi^m_{gu}(t)}{\partial t}\right|_r$$

$$u=x,y,z$$
$$v=y,z,x$$
$$w=z,x,y$$

At the left side A_{ul} both integrals give the same with a minus sign.

Fick's law finally is

$$D_g^m(t) \frac{\partial \psi_{gu}^m(t)}{\partial u}\Big|_s = (j_{gus}^{-m} - j_{gus}^{+m}) \qquad (3.7)$$

$u=x,y,z$
$s=l,r$

where $\psi_{gu}^s(u,t)$ describes the average spatial dependence of the flux inside a node in direction u.

$$\psi_{gus}^m = \frac{1}{A_u} \int_{A_u} \phi_g(\bar{r},t)dvdw$$

is then the average surface flux.

Conventional nodal methods define spatial coupling coefficients by the relationship

$$C_{gu}^{m,r} = j_{gur}^{+m}/(\Delta u \cdot \phi_g^m)$$

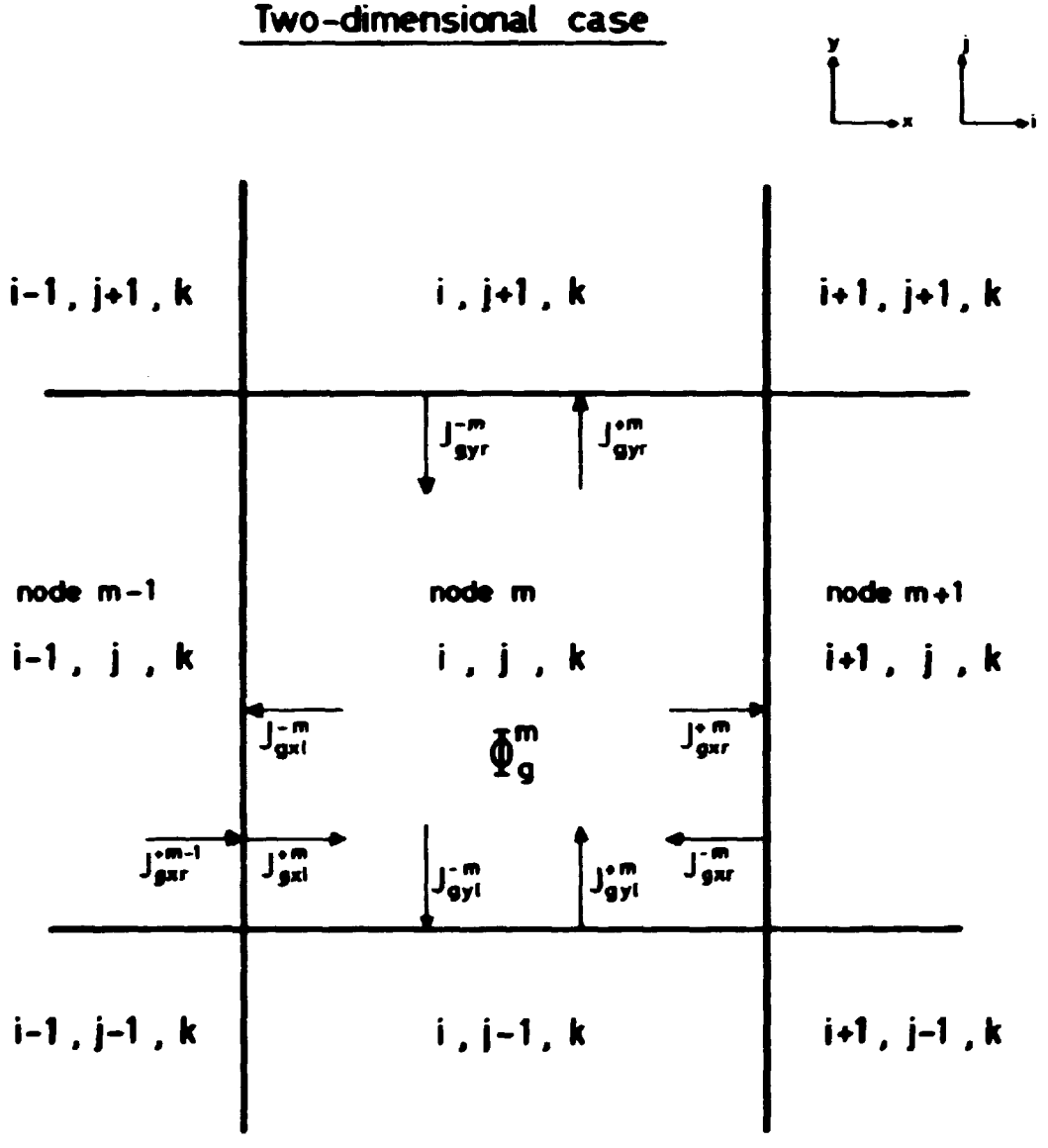
$$C_{gu}^{m,l} = j_{gul}^{-m}/(\Delta u \cdot \phi_g^m) \qquad u=x,y,z$$

If these coupling coefficients can be determined by subsidiary calculations and/or by applying subsidiary procedures Eq. (3.7) is superfluous.

Another way of dealing with the problem is to determine the spatial dependence of $\psi_{gu}^m(u,t)$ by concomitant auxiliary calculations. If $\psi_{gu}^m(u,t)$ is known, the partial currents can be determined by Eq. (3.7) in connection with partial current continuity. (What comes out of one node must go into the next.)

$$j_{gur}^{+m} = j_{gul}^{+m+1}$$

$$j_{gur}^{-m} = j_{gul}^{-m+1} \qquad u=x,y,z \qquad (3.8)$$



Partial current continuity between adjacent nodes ex.

$$J^{+m-1}_{gxr} = J^{+m}_{gxl}$$

Fig. 3.1. The basic variables for an arbitrary node m, and relations with adjacent nodes.

The one-dimensional flux $\psi^m_{gu}(u,t)$ can be expanded into a polynomial

$$\psi^m_{gu}(u) = \sum_{n=0}^{N} c^m_{gn} \cdot P_n \left(\frac{u}{\Delta u}\right) \tag{3.9}$$

In the basic variant of the nodal expansion method ψ is expanded to second order. The expansion functions that are used are

$$P_0(u) = 1$$

$$P_1(u) = u$$

$$P_2(u) = u^2 - \frac{1}{12}$$

(See moreover Appendix A.)

To satisfy the normalization condition

$$\frac{1}{\Delta u} \int_{\Delta u} \psi^m_{gu} du = \phi^m_g$$

and for the boundary conditions ψ^m_{gur} and ψ^m_{gul} the three coefficients are determined to

$$c^m_{go} = \phi^m_g$$

$$c^m_{g1} = \psi^m_{gur} - \psi^m_{gul} \tag{3.10}$$

$$c^m_{g2} = 3(\psi^m_{gur} + \psi^m_{gul} - 2\phi^m_g)$$

The diffusion theory gives an expression for the average surface flux

$$\psi^m_{gus} = 2(j^{+m}_{gus} + j^{-m}_{gus}) \tag{3.11}$$

and the insertion of the expansion (3.9) into Fick's law (3.7) results in a set of nodal equations which can be solved iteratively.

By inserting Eqs. (3.11) and (3.10) into (3.9) the following expression can be derived:

$$\phi^{m}_{gu}\left(\frac{u}{\Delta u}\right) = \phi^{m}_{g} + 2\left[j^{+m}_{gur} + j^{-m}_{gur} - \left(j^{+m}_{gul} + j^{-m}_{gul}\right)\right] \cdot \frac{u}{\Delta u}$$

$$+ 6\left(j^{+m}_{gur} + j^{-m}_{gur} + j^{+m}_{gul} + j^{-m}_{gul} - \phi^{m}_{g}\right)\left[\left(\frac{u}{\Delta u}\right)^2 - \frac{1}{12}\right]$$

This expression for the one-dimensional flux in direction u can now be inserted into Fick's law (3.7). First, the expression at the left edge of a node (s=l, $u = -\frac{1}{2}\Delta u$):

$$\frac{D^{m}_{g}}{\Delta u}\left\{2\left[j^{+m}_{gur} + j^{-m}_{gur} - \left(j^{+m}_{gul} + j^{-m}_{gul}\right)\right]\right.$$

$$\left. + 6\left(j^{+m}_{gur} + j^{-m}_{gur} + j^{+m}_{gul} + j^{-m}_{gul} - \phi^{m}_{g}\right)\right\} = j^{-m}_{gul} - j^{+m}_{gul}$$

<=>

$$\left(1+8\frac{D^{m}_{g}}{\Delta u}\right)j^{-m}_{gul} + 4\frac{D^{m}_{g}}{\Delta u}j^{+m}_{gur} = \left(1-8\frac{D^{m}_{g}}{\Delta u}\right)j^{+m}_{gul} - 4\frac{D^{m}_{g}}{\Delta u}j^{-m}_{gur} + 6\frac{D^{m}_{g}}{\Delta u}\phi^{m}_{g} \qquad (3.12)$$

Correspondingly, at the right edge of a node (s=r, $u = +\frac{1}{2}\Delta u$):

$$\frac{D^{m}_{g}}{\Delta u}\left\{2\left[j^{+m}_{gur} + j^{-m}_{gur} - \left(j^{+m}_{gul} + j^{-m}_{gul}\right)\right]\right.$$

$$\left. + 6\left(j^{+m}_{gur} + j^{-m}_{gur} + j^{+m}_{gul} + j^{-m}_{gul} - \phi^{m}_{g}\right)\right\} = j^{-m}_{gur} - j^{+}_{gur}$$

<=>

$$4\frac{D^{m}_{g}}{\Delta u}j^{-m}_{gul} + \left(1+8\frac{D^{m}_{g}}{\Delta u}\right)j^{+m}_{gur} = -4\frac{D^{m}_{g}}{\Delta u}j^{+m}_{gul} + \left(1-8\frac{D^{m}_{g}}{\Delta u}\right)j^{-m}_{gur} + 6\frac{D^{m}_{g}}{\Delta u}\phi^{m}_{g} \qquad (3.13)$$

Equations (3.12) and (3.13) can be solved with regard to the outgoing currents j^{-m}_{gul} and j^{+m}_{gul}. The result is

$$j^{-m}_{gul} = \frac{6a}{1+12a}\,\phi^m_g + \frac{1-48a^2}{1+16a+48a^2}\,j^{+m}_{gul} + \frac{-8a}{1+16a+48a^2}\,j^{-m}_{gur}$$

and

$$j^{+m}_{gur} = \frac{6a}{1+12a}\,\phi^m_g + \frac{-8a}{1+16a+48a^2}\,j^{+m}_{gul} + \frac{1-48a^2}{1+16a+48a^2}\,j^{-m}_{gur}$$

where $a = \dfrac{D^m_g}{\Delta u}$

In a shorter notation, with the introduction of the A-coefficient the outgoing currents are given as

$$j^{-m}_{gul} = A^m_{ogu}\,\phi^m_g + A^m_{1gu}\,j^{+m}_{gul} + A^m_{2gu}\,j^{-m}_{gur}$$

$$j^{+m}_{gur} = A^m_{ogu}\,\phi^m_g + A^m_{2gu}\,j^{+m}_{gul} + A^m_{1gu}\,j^{-m}_{gur} \qquad (3.14)$$

where

$$A^m_{ogu} = \frac{6a}{1+12a}$$

$$A^m_{1gu} = \frac{1-48a^2}{1+16a+48a^2}$$

$$A^m_{2gu} = \frac{-8a}{1+16a+48a^2}$$

The expression for the outgoing currents (3.14) can now be inserted into the nodal equation, (3.5), to yield the final nodal balance equation

$$\frac{1}{v_g}\frac{d}{dt}\phi_g^m = \sum_{u=x,y,z}\frac{1}{\Delta u}\left[2A_{ogu}^m\phi_g^m + (A_{1gu}^m+A_{2gu}^m-1)(j_{gul}^{+m}+j_{gur}^{-m})\right]$$

$$+ \Sigma_{rg}^m\phi_g^m = \sum_{g'=1}^{G}\left[\Sigma_{gg'}^m+(1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma_{fg'}^m\right]\phi_{g'}^m + \sum_{i=1}^{I}\lambda_i\chi_{dg}^i C_i^m$$

Rearranging

$$\left[\frac{1}{v_g}\frac{d}{dt} + \sum_{u=x,y,z}\frac{2A_{ogu}^m}{\Delta u} + \Sigma_{rg}^m\right]\phi_g^m = \sum_{g'=1}^{G}\left[\Sigma_{gg'}^m+(1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma_{fg'}^m\right]\phi_{g'}^m,$$

(3.15)

$$+ \sum_{i=1}^{I}\lambda_i\chi_{dg}^i C_i^m + \sum_{u=x,y,z}\frac{1}{\Delta u}(1 - A_{1gu}^m - A_{2gu}^m)(j_{gul}^{+m} + j_{gur}^{-m})$$

The A's are given by

$$A_{ogu}^m = \frac{6a}{1+12a}$$

$$1 - A_{1gu}^m - A_{2gu}^m = \frac{24a}{1+12a} = 4 \cdot A_{ogu}^m$$

Now a iteration procedure can be outlined for the basic variant of the nodal expansion method (NEM). From the preceding iteration step all average fluxes and outgoing currents are known. The incoming currents can be found by the continuity condition, Eq. (3.8). At the boundaries of the core the incoming currents can be found by using some albedos.

In the general case an albedo-matrix is used with the possibility of down- (and up-) scattering in the reflector

$$J_{in} = \alpha \cdot J_{out}$$

Formally, the iteration procedure is:

1) Solve

$$\frac{1}{v}\frac{d\Phi}{dt} = A\,\Phi + B\,J_{in} + E \cdot C$$

$$\frac{dC}{dt} = -\lambda C + E\,\Phi \qquad (3.16)$$

with regard to Φ and C for all nodes.

2) Calculate the outgoing currents

$$J_{out} = G\,\Phi + H \cdot J_{in} \qquad (3.17)$$

3) Calculate new incoming currents with the continuity condition (3.8)

$$J_{in} = P \cdot J_{out}$$

3.2. Stationary equations

The stationary equations are found by setting all time derivatives equal to zero and by neglecting the delayed neutrons.

Equation (3.15) now becomes

$$\left[\sum_{u=x,y,z} \frac{2A^m_{ogu}}{\Delta u} + \Sigma^m_{rg}\right]\phi^m_g = \sum_{g'=1}^{G} \left[\Sigma^m_{gg'} + \frac{\chi_{pg}}{\lambda}\,\nu\Sigma^m_{fg'}\right]\phi^m_{g'}$$

$$+\sum_{u=x,y,z} \frac{1}{\Delta u}\left(1 - A^m_{1gu} - A^m_{2gu}\right)\left(j^{+m}_{gul} + j^{-m}_{gur}\right)$$

The other equations, (3.14) and (3.8), are unchanged.

In the two-group theory the formal equations are

$$A\,\Phi = B\,J^{in} \tag{3.18}$$

$$J^{out} = D\,\Phi + E\,J^{in} \tag{3.19}$$

where (3.18) are two equations with two unknowns per node which easily can be solved and (3.19) are 12 equations. The continuity equations can be written

$$J^{in} = P\,J^{out}$$

where P is some kind of permutation matrix.

3.3. Higher-order approximations

The basic variant of NEM with an expansion of Eq. (3.9) to second order is unfortunately not very accurate and therefore the higher-order coefficients (third and fourth order) have to be determined by additional calculations. The higher-order coefficients can be determined by requiring that (3.9) solves the one-dimensional diffusion equation in a weighted residual sense. The one-dimensional equation in node m, energy group g, and coordinate u, which are equivalent to Eq. 3.1 is

$$\frac{1}{v_g}\frac{\partial \psi^m_{gu}}{\partial t} - \frac{\partial}{\partial u} D^m_g \frac{\partial}{\partial u} \psi^m_{gu} + (\Sigma^m_{ag} + \Sigma^m_{sg})\psi^m_{gu} \tag{3.20}$$

$$= \sum_{g'=1}^{G} \left[\Sigma^m_{gg'} + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma^m_{fg'}\right]\psi^m_{g'u} + \sum_{i=1}^{I}\lambda_i \chi^i_{dg}\,\xi^m_{iu} - L^m_{gu}$$

where

ψ^m_{gu} is the one-dimensional flux in node m, energy group g and coordinate u.

ξ^m_{iu} is the solution of the partially integrated precursor equation

$$\frac{\partial \xi^m_{iu}}{\partial t} + \lambda_i \xi^m_{iu} = \frac{1}{\lambda} \sum_{g'=1}^{G} \beta_i \nu \Sigma^m_{fg'} \psi^m_{g'u} \qquad (3.21)$$

L^m_{gu} is the transverse leakage out of node m in the directions v and w

$$L^m_{gu} = -\frac{1}{A_u} \int_{-1/2\Delta v}^{+1/2\Delta v} \int_{-1/2\Delta w}^{+1/2\Delta w} \left[\frac{\partial}{\partial v} D^m_g \frac{\partial}{\partial v} + \frac{\partial}{\partial w} D^m_g \frac{\partial}{\partial w}\right] \phi_g(\bar{r},t)\,dv\,dw \qquad (3.22)$$

An approximation for the transverse leakage L^m_{gu} can be obtained by using leakage information from adjacent nodes, and a parabolic representation has proven to be sufficiently accurate

$$L^m_{gu} = \sum_{n=0}^{2} L^m_{gn} \cdot P\left(\frac{u}{\Delta u}\right) \qquad (3.23)$$

The coefficients are

$$L^m_{go} = \bar{L}^m_{gu} = \frac{J^{out}_v}{\Delta v} + \frac{j^{out}_w}{\Delta w} - \frac{j^{in}_v}{\Delta v} - \frac{j^{in}_w}{\Delta w}$$

$$= \frac{j^{-m}_{gv1}+j^{+m}_{gvr}}{\Delta v} + \frac{j^{-m}_{gw1}+j^{+m}_{gwr}}{\Delta w}$$

$$- \frac{j^{+m}_{gv1}+j^{-m}_{gvr}}{\Delta v} - \frac{j^{+m}_{gw1}+j^{-m}_{gwr}}{\Delta w}$$

$$L^m_{g1} = L^m_{gur} - L^m_{gul}$$

$$L^m_{g2} = 3[(L^m_{gur} + L^m_{gul}) - 2\bar{L}^m_{gu}]$$

where $\bar{L}^m_{gu}$ is the average transverse leakage for node m and direction u.

The boundary values L^m_{gur} and L^m_{gul} can be derived from continuity conditions at the boundaries of each node

$$L^{m-1}_{gur} = L^m_{gul}$$

$$D^{m-1}_g \frac{\partial}{\partial u} L^{m-1}_{gur} = D^m_g \frac{\partial}{\partial u} L^m_{gul} \qquad (3.24)$$

where m-1 and m are numbers of adjacent nodes. Equations (3.24) forms a tridiagonal system for the determination of the boundary values as functions of the average transverse leakages

$$\bar{L}^m_{gu}.$$

To save calculations a linear approximation which fulfils Eq. (3.24) is used.

$$L^m_{gul} = (D^m_g \bar{L}^m_{gu} + D^{m-1}_g \bar{L}^{m-1}_{gu})/(D^m_g + D^{m-1}_g)$$

and (3.25)

$$L^m_{gur} = (D^m_g \bar{L}^m_{gu} + D^{m+1}_g \bar{L}^{m+1}_{gu})/(D^m_g + D^{m+1}_g)$$

If node m is at the boundary of the core

$$\bar{L}^{m-1}_{gu} \text{ or } \bar{L}^{m+1}_{gu} \quad ,$$

the transverse leakage in the so-called vacuum node is set to zero. It does not give significantly different results by using a zero transverse leakage value at the boundary instead of the above method.

If we first look at the stationary problem the one-dimensional diffusion equation, (3.20), reduces to

$$-D^m_g \frac{\partial^2 \psi^m_{gu}}{\partial u^2} + [\Sigma^m_{ag}+\Sigma^m_{sg}]\psi^m_{gu} - \sum_{g'=1}^{G} [\Sigma^m_{gg'} + \frac{\chi_{pg}}{\lambda} \nu\Sigma^m_{fg'}]\psi^m_{g'u}$$

$$+ L^m_{gu} = 0 \qquad (3.26)$$

Equation (3.26) is solved by weighted residual technique:

$$\int_{-\frac{1}{2}\Delta u}^{+\frac{1}{2}\Delta u} w_i(u) \cdot \{-D^m_g \frac{\partial^2 \psi^m_{gu}}{\partial u^2} + [\Sigma^m_{ag} + \Sigma^m_{sg}]\psi^m_{gu}$$

$$-\sum_{g'=1}^{G} [\Sigma^m_{gg'} + \frac{\chi_{pg}}{\lambda} \nu\Sigma^m_{fg'}]\psi^m_{g'u} + L^m_{gu}\} \, du = 0 \qquad (3.27)$$

The weight functions used in the weighted residual formula are the same as the expansion functions for the one-dimensional flux

$$w_1(u) = P_1(u) = u$$

$$w_2(u) = P_2(u) = u^2 - \frac{1}{12}$$

The one-dimensional flux is expanded by (3.9) to N'th order and the transverse leakage is given by (3.23).

With these terms inserted in the residual equation we obtain

$$\int_{-\frac{1}{2}\Delta u}^{+\frac{1}{2}\Delta u} w_i(u)\cdot\{-D_g^m \sum_{n=o}^{N} c_{gnu}^m \frac{d^2P_n(u)}{du^2} + \Sigma_{rg}^m \cdot \sum_{n=o}^{N} c_{gnu}^m P_n(u)$$

$$-\sum_{g'=1}^{G} [\Sigma_{gg'}^m + \frac{\chi_{pg}}{\lambda} \nu\Sigma_{fg'}^m] \cdot \sum_{n=o}^{N} c_{g'nu}^m P_n(u)$$

$$+\sum_{n=o}^{2} L_{gn}^m P_n(u)\}du = 0 \qquad (3.28)$$

The only u-dependent variables in (3.28) are products of the types

$$P_i(u)\cdot P_j(u) \quad \text{and} \quad P_i(u) \cdot \frac{d^2P_j(u)}{du^2}$$

and the integrals of these products are calculated in Appendix A for various values of i and j.

With the weight function $w_1 = P_1(u)$ and maximum order of four (N=4) Eq. (3.28) gives the equation system

$$- \frac{D_g^m}{\Delta u^2} \cdot \frac{1}{2} c_{g3u}^m + \Sigma_{rg}^m \left(\frac{1}{12} c_{g1u}^m - \frac{1}{120} c_{g3u}^m\right)$$

$$- \sum_{g'=1}^{G} \left(\Sigma_{gg'}^m + \frac{\chi_{pg}}{\lambda} \nu\Sigma_{fg'}^m\right) \cdot \left(\frac{1}{12} c_{g'1u}^m - \frac{1}{120} c_{g'3u}^m\right) + \frac{1}{12} L_{g1}^m = 0$$

$$g = 1,2, \ldots, G.$$

The equation system is equivalent to

$$\begin{aligned}
&K_{11} \cdot c_{13u}^m + K_{12} \cdot c_{23u}^m + \ldots + K_{1G} \cdot c_{G3u}^m = B_1 \\
&K_{21} \cdot c_{13u}^m + K_{22} \cdot c_{23u}^m + \ldots + K_{2G} \cdot c_{G3u}^m = B_2 \qquad (3.29) \\
&\quad - \\
&\quad - \\
&\quad - \\
&K_{G1} \cdot c_{13u}^m + K_{G2} \cdot c_{23u}^m + \ldots + K_{GG} \cdot c_{G3u}^m = B_G
\end{aligned}$$

where

$$K_{ii} = - \frac{1}{2} \frac{D_i^m}{\Delta u^2} - \frac{1}{120} \Sigma_{ri}^m + \frac{1}{120} \left(\Sigma_{ii}^m + \frac{\chi_{pi}}{\lambda} \nu\Sigma_{fi}^m\right)$$

$$K_{ij} = \frac{1}{120} \left(\Sigma_{ij}^m + \frac{\chi_{pi}}{\lambda} \nu\Sigma_{fj}^m\right) \qquad i \neq j$$

$$B_i = - \frac{1}{12} \Sigma_{ri}^m c_{i1u}^m + \frac{1}{12} \sum_{g'=1}^{G} \left(\Sigma_{ig'}^m + \frac{\chi_{pi}}{\lambda} \nu\Sigma_{fg'}^m\right) \cdot c_{g'1u}^m - \frac{1}{12} L_{i1}^m$$

With the weight function $w_2 = P_2(u)$ and maximum order of four we obtain

$$-\frac{D_g^m}{\Delta u^2}\cdot\frac{1}{15}\,c_{g4u}^m+\Sigma_{rg}^m\left(\frac{1}{180}\,c_{g2u}^m-\frac{1}{2100}\,c_{g4u}^m\right)$$

$$-\sum_{g'=1}^{G}\left(\Sigma_{gg'}^m+\frac{\chi_{pg}}{\lambda}\,\nu\Sigma_{fg'}^m\right)\left(\frac{1}{180}\,c_{g'2u}^m-\frac{1}{2100}\,c_{g'4u}^m\right)+\frac{1}{180}\,L_{g2}^m=0$$

$$g = 1,2,\ldots,G$$

It gives the equation system

$$\begin{aligned}
&K_{11}\cdot c_{14u}^m+K_{12}\cdot c_{24u}^m+\ldots+K_{1G}\cdot c_{G4u}^m=B_1\\
&K_{21}\cdot c_{14u}^m+K_{22}\cdot c_{24u}^m+\ldots+K_{2G}\cdot c_{G4u}^m=B_2\\
&-\\
&-\\
&K_{G1}\cdot c_{14u}^m+K_{G2}\cdot c_{24u}^m+\ldots+K_{GG}\cdot c_{G4u}^m=B_G
\end{aligned}\qquad(3.30)$$

with

$$K_{ii}=-\frac{1}{15}\,\frac{D_i^m}{\Delta u^2}-\frac{1}{2100}\,\Sigma_{ri}^m+\frac{1}{2100}\left(\Sigma_{ii}^m+\frac{\chi_{pi}}{\lambda}\,\nu\Sigma_{fi}^m\right)$$

$$K_{ij}=\frac{1}{2100}\left(\Sigma_{ij}^m+\frac{\chi_{pi}}{\lambda}\,\nu\Sigma_{fj}^m\right)\qquad i\neq j$$

$$B_i=-\frac{1}{180}\,\Sigma_{ri}^m c_{12u}^m+\frac{1}{180}\sum_{g'=1}^{G}\left(\Sigma_{ig'}^m+\frac{\chi_{pi}}{\lambda}\,\nu\Sigma_{fg'}^m\right)\cdot c_{g'2u}^m-\frac{1}{180}\,L_{i2}^m$$

The two equation systems (3.29) and (3.30) can easily be solved with regard to the expansion coefficients by direct methods (Gauss elimination). In the common case with only two energy groups there are two equations with two unknowns that are to be solved - but it is in every node in each direction!

The expansion function just found together with the lower-order expansion coefficients from (3.10) can now be inserted in the one-dimensional flux expansion (3.9)

$$\psi^m_{gu}(u) = \phi^m_g + 2[j^{+m}_{gur} + j^{-m}_{gur} - (j^{+m}_{gul} + j^{-m}_{gul})]\,\frac{u}{\Delta u}$$

$$+ 6(j^{+m}_{gur} + j^{-m}_{gur} + j^{+m}_{gul} + j^{-m}_{gul} - \phi^m_g)[(\frac{u}{\Delta u})^2 - \frac{1}{12}]$$

$$+ c^m_{g3u} \cdot P_3(\frac{u}{\Delta u}) + c^m_{g4u} \cdot P_4(\frac{u}{\Delta u})$$

The expansion is inserted in Fick's law (3.7) at the left edge of the node ($s = 1$, $u = -\frac{1}{2}\Delta u$)

$$\frac{D^m_g}{\Delta u}\{2[j^{+m}_{gur} + j^{-m}_{gur} - (j^{+m}_{gul} + j^{-m}_{gul})]$$

$$- 6(j^{+m}_{gur} + j^{-m}_{gur} + j^{+m}_{gul} + j^{-m}_{gul} - \phi^m_g)$$

$$+ \frac{1}{2} c^m_{g3u} - \frac{1}{5} c^m_{g4u}\} = j^{-m}_{gul} - j^{+m}_{gul}$$

<=>

$$(1+8\frac{D^m_g}{\Delta u})j^{-m}_{gul} + 4\frac{D^m_g}{\Delta u} j^{+m}_{gur} = (1-8\frac{D^m_g}{\Delta u})j^{+m}_{gul} - 4\frac{D^m_g}{\Delta u} j^{-m}_{gur} + 6\frac{D^m_g}{\Delta u}\phi^m_g$$

$$+ \frac{D^m_g}{\Delta u}(\frac{1}{2} c^m_{g3u} - \frac{1}{5} c^m_{g4u}) \qquad (3.31)$$

Correspondingly, at the right edge of the node (s = r, u = $+\frac{1}{2}\Delta u$)

$$\frac{D_g^m}{\Delta u}\{2[j_{gur}^{+m} + j_{gur}^{-m} - (j_{gul}^{+m} + j_{gul}^{-m})]$$

$$+ 6(j_{gur}^{+m} + j_{gur}^{-m} + j_{gul}^{+m} + j_{gul}^{-m} - \phi_g^m)$$

$$+ \frac{1}{2} c_{g3u}^m + \frac{1}{5} c_{g4u}^m\} = j_{gur}^{-m} - j_{gur}^{+m}$$

<=>

$$4\frac{D_g^m}{\Delta u} j_{gul}^{-m} + (1+8\frac{D_g^m}{\Delta u}) j_{gur}^{+m} = -4\frac{D_g^m}{\Delta u} j_{gul}^{+m} + (1-8\frac{D_g^m}{\Delta u}) j_{gur}^{-m}$$

$$+ 6\frac{D_g^m}{\Delta u}\phi_g^m - \frac{D_g^m}{\Delta u}(\frac{1}{2} c_{g3u}^m + \frac{1}{5} c_{g4u}^m) \qquad (3.32)$$

Equation (3.31) and (3.32) can now be solved with regard to the outgoing currents j_{gul}^{-m} and j_{gur}^{+m}

$$j_{gul}^{-m} = A_{ogu}^m \phi_g^m + A_{1gu}^m j_{gul}^{+m} + A_{2gu}^m j_{gur}^{-m}$$

$$+ A_{3gu}^m \cdot c_{g3u}^m - A_{4gu}^m \cdot c_{g4u}^m \qquad (3.33)$$

$$j_{gur}^{+m} = A_{ogu}^m \phi_g^m + A_{2gu}^m j_{gul}^{+m} + A_{1gu}^m j_{gur}^{-m}$$

$$- A_{3gu}^m \cdot c_{g3u}^m - A_{4gu}^m \cdot c_{g4u}^m$$

With

$$\frac{D_g^m}{\Delta_u} = a$$

the above A-coefficients are

$$A_{ogu}^m = \frac{6a}{1+12a}$$

$$A_{1gu}^m = \frac{1-48a^2}{1+16a+48a^2}$$

$$A_{2gu}^m = \frac{-8a}{1+16a+48a^2} \qquad (3.34)$$

$$A_{3gu}^m = \frac{1}{2} \cdot \frac{a}{1+4a}$$

$$A_{4gu}^m = \frac{1}{5} \frac{a}{1+12a} = \frac{1}{30} A_{ogu}^m$$

The expressions for the outgoing currents (3.33) are inserted in the nodal equation, (3.5)

$$\sum_{u=x,y,z} \frac{1}{\Delta_u} \left[2A_{ogu}^m \phi_g^m - 2A_{4gu}^m \cdot c_{g4u}^m \right.$$

$$\left. + (A_{1gu}^m + A_{2gu}^m - 1)(j_{gul}^{+m} + j_{gur}^{-m}) \right]$$

$$+ \Sigma_{rg}^m \phi_g^m = \sum_{g'=1}^{G} (\Sigma_{gg'}^m + \frac{\chi_{pg}}{\lambda} \nu\Sigma_{fg'}^m) \phi_{g'}^m$$

Rearranging and using $1-A^m_{1gu}-A^m_{2gu} = 4A^m_{ogu}$

$$\left(\sum_{u=x,y,z} \frac{2A^m_{ogu}}{\Delta u} + \Sigma^m_{rg}\right)\phi^m_g = \sum_{g'=1}^{G}\left(\Sigma^m_{gg'} + \frac{\chi_{pg}}{\lambda}\nu\Sigma^m_{gg'}\right)\phi^m_{g'}$$

$$+ \sum_{u=x,y,z} \frac{1}{\Delta u}\left[4A^m_{ogu}\left(j^{+m}_{gul} + j^{-m}_{gur}\right) + \frac{1}{15}A^m_{ogu}\cdot c^m_{g4u}\right] \tag{3.35}$$

With higher-order expansion coefficients the steady-state iteration procedure is:

1. Calculate the higher-order expansion coefficients c_3 and c_4 from previous values of fluxes and currents.
2. Find the fluxes from $A \cdot \phi = B \cdot J^{in} + B_1 \cdot c_4$
3. Calculate new outgoing currents $J^{out} = D\,\phi + E\,J^{in} + E_1\,c_3 + E_2\,c_4$
4. Find the new incoming currents $J^{in} = P \cdot J^{out}$

3.4. Time-dependent theory

In the preceding paragraphs I have given a very detailed description of the theory for the steady-state calculations with the nodal expansion method. The theory for the time-dependent problems is the same and the resulting equations are identical with the exception of a few extra terms from the time-derivative of the flux and from the delayed neutrons.

The expressions for the outgoing currents as function of incoming currents and average fluxes (3.33) are still valid for dynamic problems. The nodal equation is obtained by combining Eqs. (3.15) and (3.35)

$$\left(\frac{1}{v_g}\frac{d}{dt} + \sum_{u=x,y,z}\frac{2A^m_{ogu}}{\Delta u} + \Sigma^m_{rg}\right)\phi^m_g = \sum_{g'=1}^{G}\left[\Sigma^m_{gg'} + (1-\beta)\frac{\chi_{pg}}{\lambda}\nu\Sigma^m_{fg'}\right]\phi^m_{g'}$$

$$+ \sum_{i=1}^{I}\lambda_i\chi^i_{dg}C^m_i + \sum_{u=x,y,z}\frac{1}{\Delta u}\left[4A^m_{ogu}\left(j^{+m}_{gul} + j^{-m}_{gur}\right)\right.$$

$$\left. + \frac{1}{15}A^m_{ogu}\cdot c^m_{g4u}\right] \qquad (3.36)$$

The time-derivative of the flux is approximated with a simple first-order backward difference formula

$$\frac{d\phi^m_g(t)}{dt} = \frac{1}{\Delta t}\left[\phi^m_g(t) - \phi^m_g(t-\Delta t)\right] \qquad (3.37)$$

which is inserted in (3.36). The term $\phi^m_g(t-\Delta t)$ is known from the previous timestep.

The contribution from the delayed neutrons has to be calculated by the precursor equation, (3.6). The solution of the precursor equation is approximated by

$$C^m_i(t) = C^m_i(t-\Delta t)\cdot e^{-\lambda_i\cdot\Delta t}$$

$$+ \left[\frac{1}{\lambda}\sum_{g'=1}^{G}\beta_i\nu\Sigma^m_{fg'}\phi^m_{g'}(t)\right]\cdot\frac{1-e^{-\lambda_i\Delta t}}{\lambda_i} \qquad (3.38)$$

The one-dimensional equation for determining the higher-order coefficients (3.20) has two more terms than were included in the derivation in Section 3.3. The first term is the derivative of

the one-dimensional average flux, and it is assumed that it has the same time-dependence as the node average flux. Moreover, it is also assumed that it has the same spatial dependence as the one-dimensional flux itself.

$$\frac{1}{v_g}\frac{\partial \psi^m_{gu}}{\partial t} = \frac{1}{v_g}\cdot\frac{1}{\phi^m_g}\frac{d\phi^m_g}{dt}\cdot\psi^m_{gu} \tag{3.39}$$

The time-derivative of the average flux is approximated by Eq. (3.37).

The second term which has to be added is the contribution from the delayed neutrons. The one-dimensional precursor Eq (3.21) has to be solved and it could also be done by weighted residual techniques. Instead of attempting this rather hard work, it is assumed that the spatial distribution of the delayed neutrons follows the flux distribution.

$$\xi^m_{iu} = \frac{\sum_{g'=1}^{G}\psi^m_{g'u}}{\sum_{g'=1}^{G}\phi^m_{g'}}\cdot C^m_i \tag{3.40}$$

and that it suffices to expand the local one-dimensional flux to second order.

With these additional terms the one-dimensional equation can be solved by weighted residual techniques in quite the same way as in the static case. The coefficients in Eq. (3.29) become

$$K_{ii} = -\frac{1}{2}\frac{D_i^m}{\Delta u^2} - \frac{1}{120}\Sigma_{ri}^m + \frac{1}{120}\left[\Sigma_{ii}^m + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma_{fi}^m\right] - \frac{1}{120}\frac{1}{v_i}\frac{d\phi_i^m}{dt}$$

$$K_{ij} = \frac{1}{120}\left[\Sigma_{ij}^m + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma_{fj}^m\right] \qquad i \neq j$$

$$B_i = -\frac{1}{12}\Sigma_{ri}^m \cdot c_{i1u}^m + \frac{1}{12}\sum_{g'=1}^{G}\left[\Sigma_{ig'}^m + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma_{fg'}^m\right]\cdot c_{g'1u}^m$$

$$- \frac{1}{12}L_{i1}^m - \frac{1}{12}\frac{1}{v_i}\frac{d\phi_i^m}{dt} \cdot c_{i1u}^m$$

$$+ \frac{1}{12}\chi_{di}\sum_{k=1}^{I} \cdot \frac{\sum_{g'=1}^{G} c_{g'1u}^m}{\sum_{g'=1}^{G}\phi_{g'}^m} \cdot \lambda_k \cdot C_k^m$$

and in Eq. (3.30)

$$K_{ii} = -\frac{1}{15}\frac{D_i^m}{\Delta_u^2} - \frac{1}{2100}\Sigma_{ri}^m + \frac{1}{2100}\left[\Sigma_{ii}^m + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma_{fi}^m\right]$$

$$- \frac{1}{2100}\frac{1}{v_i}\frac{d\phi_i^m}{dt}$$

$$K_{ij} = \frac{1}{2100}\left[\Sigma^m_{ij} + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma^m_{fj}\right] \qquad i \neq j$$

$$B_i = -\frac{1}{180}\Sigma^m_{ri}\cdot c^m_{i2u} + \frac{1}{180}\sum_{g'=1}^{G}\left[\Sigma^m_{ig'} + (1-\beta)\frac{\chi_{pi}}{\lambda}\nu\Sigma^m_{fg'}\right]\cdot c^m_{g'2u}$$

$$-\frac{1}{180}L^m_{i2} - \frac{1}{180}\frac{1}{v_i}\frac{d\phi^m_i}{dt}\cdot c^m_{i2u}$$

$$+\frac{1}{180}\chi_{di}\sum_{k=1}^{I}\cdot\frac{\sum_{g'=1}^{G} c^m_{g'2u}}{\sum_{g'=1}^{G}\phi^m_{g'}}\cdot\lambda_k\cdot C^m_k$$

The general dynamic iteration scheme can now be outlined:

1. Calculate the higher-order expansion coefficients c_3 and c_4. from previous values of fluxes, currents and delayed neutrons.
2. Calculate the fluxes from
   $A\cdot\Phi(t) = B\cdot J^{in}+B_1\cdot c_4+B_2\cdot\Phi(t-\Delta t)+B_3\cdot C.$
3. Calculate new outgoing currents
   $J^{out} = D\cdot\Phi(t)+E\cdot J^{in}+E_1\cdot c_3+E_2\cdot c_4.$
4. Calculate new incoming currents
   $J^{in} = p\cdot J^{out}.$
5. Calculate new values of the delayed neutrons
   $C(t) = F\cdot C(t-\Delta t)+G\cdot\Phi(t).$

4. PROGRAMMING

The nodal expansion method has been programmed in NEM, a program divided in two parts, one for steady-state calculations and the other for dynamic calculations. Before the dynamic part can be run a steady-state calculation has to be made with a dump of all variables to file as last output. Then the dynamic calculations start with a restart from the dump file.

Each program consists of a main part and several subroutines as shown on the structure diagrams in Figs. 4.1 and 4.2. In the following the main parts of the program will be described in more detail. The program structure is straightforward and it should make it easy to combine with a hydraulic program.

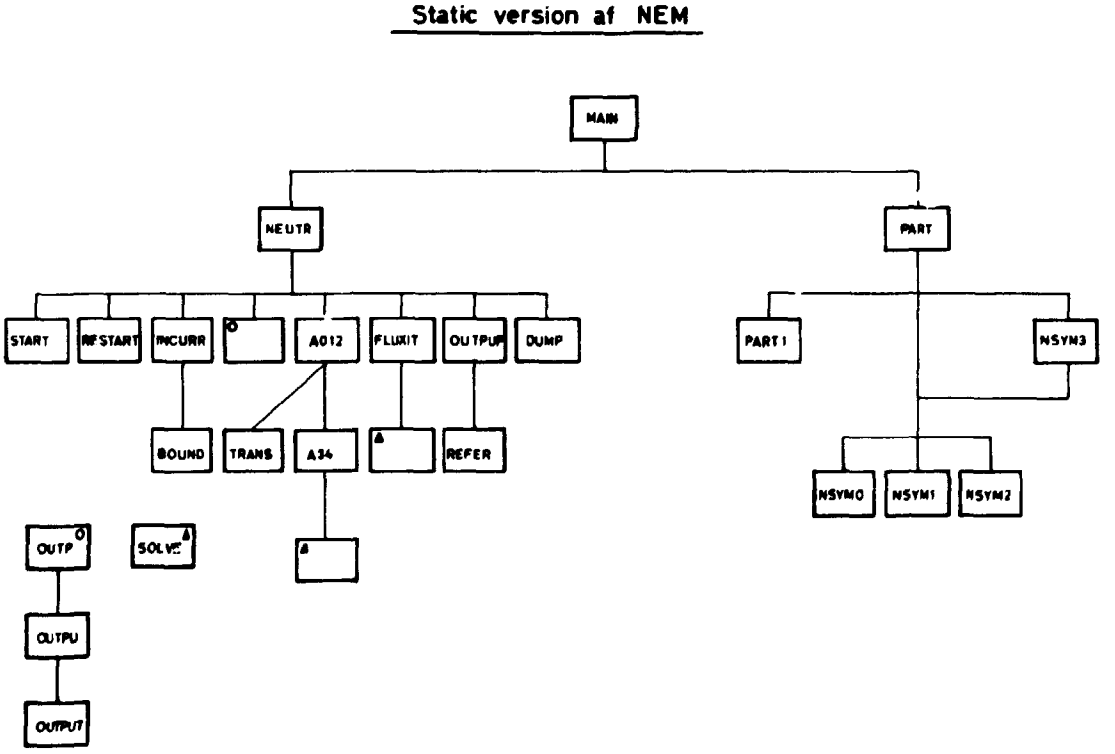


Fig. 4.1. Structure diagram for the static part of the NEM-code.

Dynamic version af NEM

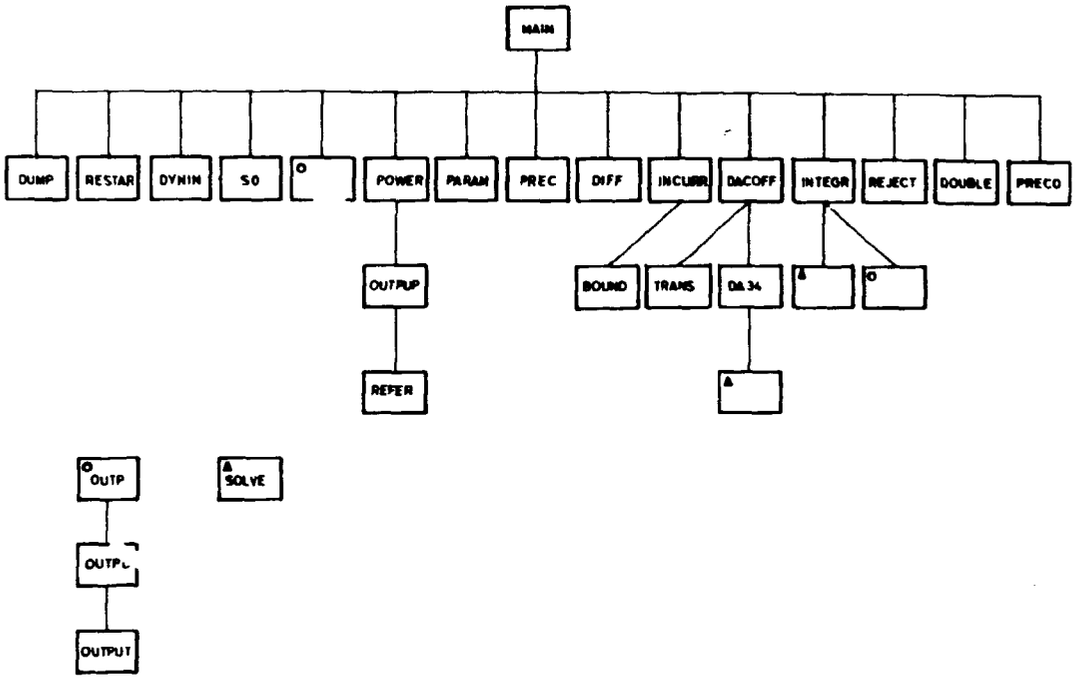


Fig. 4.2. Structure diagram for the dynamic part of the NEM-code.

4.1. The steady-state main program

The main program consists all file declarations and all global array declarations. The files used are described in the input description in Appendix D.

The arrays, which have to be global are shown beneath. The array dimensions should be equal to or greater than the values given in the table. The letters used in the dimension sizes have the following meaning:

G	number of energy groups.
I	number of nodes in the x-direction.
J	number of nodes in the y-direction.
K	number of nodes in the z-direction.
NAPP	the approximation order + 1.
NNODE	number of different fuel types.

Table 4.1

List of global arrays in the steady-state program

Name	Dimensions	Content
FI	G,K,J,I	Average fluxes in each node and energy group.
CXL	G,K,J,I	Average partial currents out of each node in the x-direction at the left edge.
CXR	G,K,J,I	Average partial currents out of each node in the x-direction at the right edge.
CYL	G,K,J,I	Average partial currents out of each node in the y-direction at the left edge.
CYR	G,K,J,I	Average partial currents out of each node in the y-direction at the right edge.
CZL	G,K,J,I	Average partial currents out of each node in the z-direction at the left edge.
CZR	G,K,J,I	Average partial currents out of each node in the y-direction at the right edge.
CINXL	G,K,J,I	Average partial currents into each node in the x-direction at the left edge.
CINXR	G,K,J,I	Average partial currents into each node in the x-direction at the right edge.
CINYL	G,K,J,I	Average partial currents into each node in the y-direction at the left edge.
CINYR	G,K,J,I	Average partial currents into each node in the y-direction at the right edge.
CINZL	G,K,J,I	Average partial currents into each node in the z-direction at the left edge.

Name	Dimensions	Content
CINZR	G,K,J,I	Average partial currents into each node in the z-direction at the right edge.
P	K,J,I	Average power in each node.
AX	NAPP,G,K,J,I	The A-coefficients in each node and energy group in the x-direction.
AY	NAPP,G,K,J,I	The A-coefficients in each node and energy group in the y-direction.
AZ	NAPP,G,K,J,I	The A-coefficients in each node and energy group in the z-direction.
NODE	K,J,I	Fuel type in each node.
D	G,NNODE	The diffusion coefficients for each fuel type.
SIGA	G,NNODE	The absorption cross sections for each fuel type.
SIGG	G,G,NNODE	The scattering cross sections for each fuel type.
USIG	G,NNODE	The product of ν and the fission cross sections for each fuel type.
KHI	G	The emission spectrum.
AA	G,G	Auxiliary array with the calculated coefficients for the nodal equation system.
BB	G	Auxiliary array with the calculated right side of the nodal equation system.
IPVT	G	Auxiliary array with the pivot elements for the nodal equation system.
IMIN	J	The first node in the x-direction in each row used in the geometrical description of the reactor.
IMAX	J	The last node in the x-direction in each row.
JMIN	I	The first node in the y-direction in each column.
JMAX	I	The last node in the y-direction in each column.

Name	Dimensions	Content
WEIGHT	J,I	Weight factors for each channel.
ALBXL	G,G,J	The albedo-matrix in the x-direction in each row at the left edge.
ALBXR	G,G,J	The albedo-matrix in the x-direction in each row at the right edge.
ALBYL	G,G,I	The albedo-matrix in the y-direction in each column at the left edge.
ALBYR	G,G,I	The albedo-matrix in the y-direction in each column at the right edge.
ALBZL	G,G	The albedo-matrix in the z-direction at the left edge.
ALBZR	G,G	The albedo-matrix in the z-direction at the right edge.
BUCK	G	The energy-dependent buckling for two-dimensional problems.
TEXT	13	Headline for the problem.
LOX	G,K,J,I	The average transverse leakage in the x-direction.
LOY	G,K,J,I	The average transverse leakage in the y-direction.
LOZ	G,K,J,I	The average transverse leakage in the z-direction.
L1X	G,K,J,I	The first expansion coefficient for the transverse leakage expansion in the x-direction.
L1Y	G,K,J,I	The first expansion coefficient for the transverse leakage expansion in the y-direction.
L1Z	G,K,J,I	The first expansion coefficient for the transverse leakage expansion in the z-direction.
L2X	G,K,J,I	The second expansion coefficient for the transverse leakage expansion in the x-direction.

Name	Dimensions	Content
L2Y	G,K,J,I	The second expansion coefficient for the transverse leakage expansion in the y-direction.
L2Z	G,K,J,I	The second expansion coefficient for the transverse leakage expansion in the z-direction.

The actual array dimensions (G,I,J,K,NAPP,NNODE) are read in the main program. They are used for dynamic allocation of array sizes in each subroutine, which also means that all arrays are transferred through the head of each subroutine.

A route diagram for the main program is shown in Fig. 4.3.

4.2. Steady-state calculation

The steering routine for steady-state calculations is the subroutine NEUTR. A flow-chart for the routine is shown in Fig. 4.4.

The number of inner iterations for each outer iteration can be chosen through the input. Besides the number of inner iterations it is possible to choose an acceleration method. From Chapter 3.2 the stationary equations are

$$A \cdot \phi(r) = B \cdot J_{in}(r-1)$$
$$J_{out}(r) = D \cdot \phi(r) + E \cdot J_{in}(r-1)$$
$$J_{in}(r) = P \cdot J_{out}(r)$$

Where r is the inner iteration number. The iteration scheme for the accelerated method is

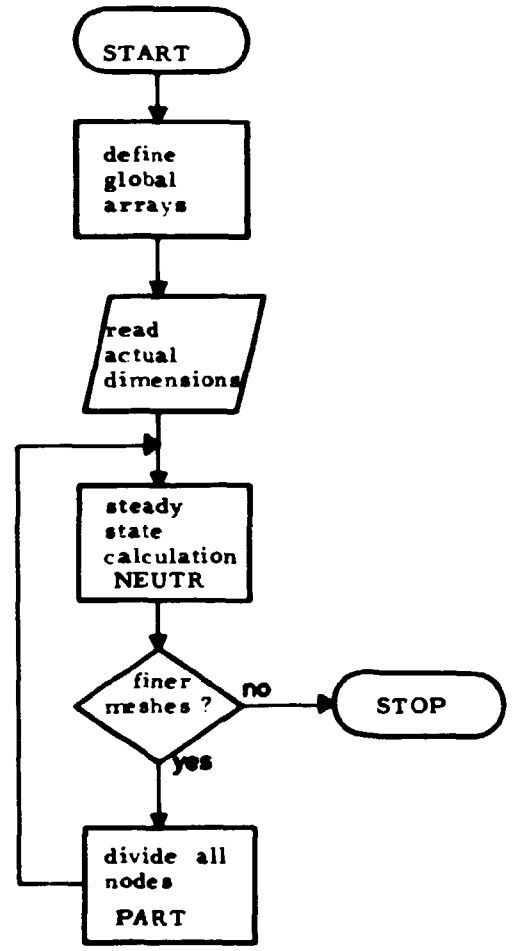


Fig. 4.3. Flow-chart for the main program for steady-state calculations.

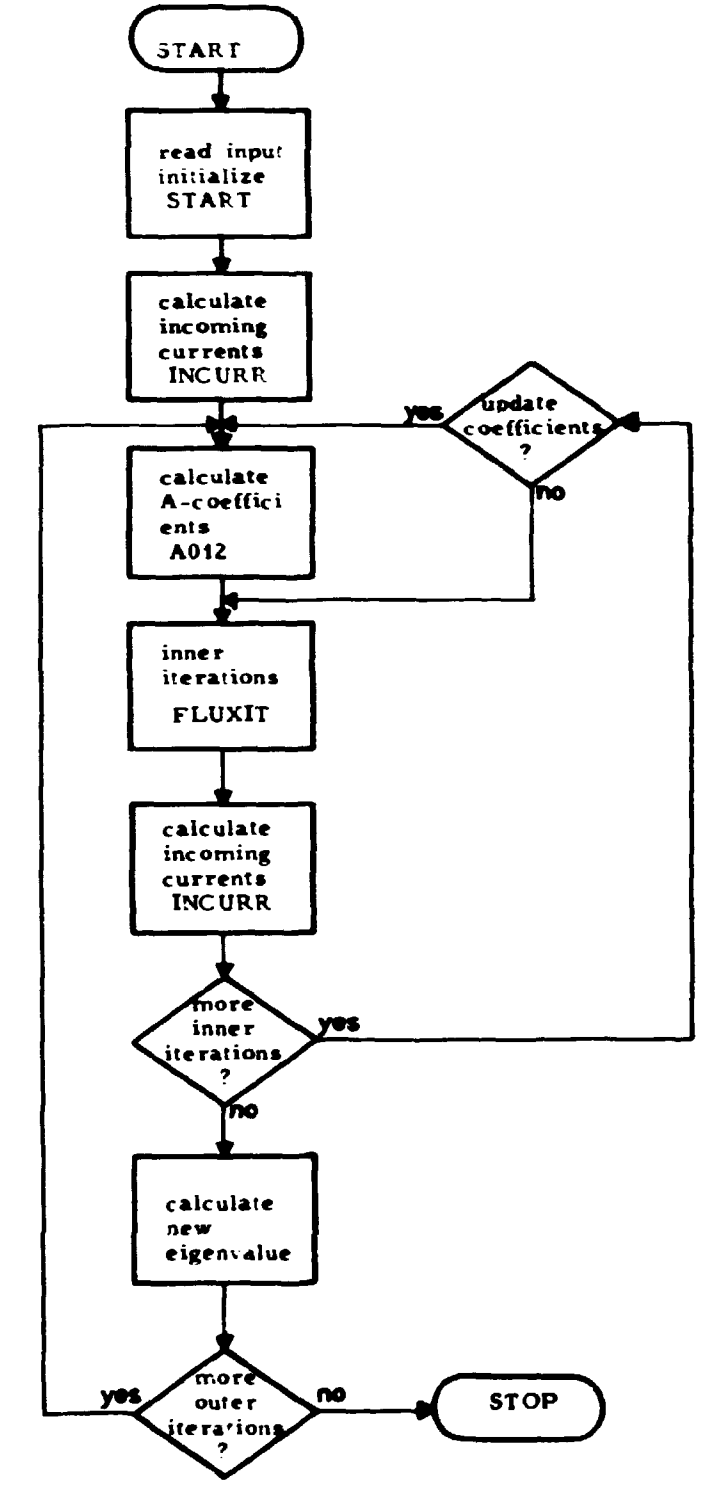


Fig. 4.4. Plow-chart for the subroutine NEUTR, which contains the steady-state steering.

1. $A \cdot \Phi(r) = B \cdot J^{in}(r-1)$

2. $J^{out}(r) = D \cdot \Phi(r) + E \cdot J^{in}(r-1)$

3.a.

$$P(r) = \frac{\sum_N \Phi_1(r)}{N} \qquad \text{for MODE} = 0,1 \tag{4.1}$$

3.b. $P(r) = \max_N \{\Phi_1(r)\}$ for MODE = 2,3

where Φ_1 is the average flux in energy group 1 in one node and N is is the total number of nodes.

The mode to be used is selected by the input.

4.a. $$\omega(r) = 1 - \gamma\left[1+\cos\left(\frac{\pi}{2\, r_{tot}} \cdot r\right)\right] \qquad \text{for r odd} \tag{4.2}$$

4.b. $$\omega(r) = 1 - \gamma\left\{1+\cos\left[\frac{\pi}{2\, r_{tot}} \cdot (2 \cdot r_{tot}+1-r)\right]\right\} \qquad \text{for r even}$$

r_{tot} is the total number of inner iterations, which has to be fixed beforehand. γ is an input constant, and a value about 0.8 is optimum in most cases.

$$5. \quad \Lambda(r) = \frac{P(r)[\Lambda(r-1) + \omega(r-1)]}{P(r-1)[1 + \omega(r-1)]} \tag{4.3}$$

$$6. \quad J^{in}(r) = \frac{1}{\Lambda(r)+\omega(r)} \cdot [P \cdot J^{out}(r) + \omega(r) \cdot J^{in}(r-1)] \tag{4.4}$$

This iteration scheme also normalizes the flux in each inner iteration. The normalization can be done in two ways depending on MODE, either on the average fluxes in group 1 or on the

maximum flux in group 1 (see step 3.a and 3.b). The best method to use depends on the actual problem, but a normalization based on the maximum flux (MODE = 2 or 3) is slightly faster than the sum-method.

Unfortunately, there is only reliable convergence in the above-described method for low local flux expansion order (second order). Therefore, it is possible to make the iterations without acceleration, but still with normalization. If MODE = 0 or 2 the parameter ω is set to zero in all iterations.

The total number of inner iterations per outer is given through the input, but besides this a tolerance on the flux is given, DF. In the inner iterations it is tested if

$$|\phi(r) - \phi(r-1)| < DF \qquad (4.5)$$

for all nodes and energy groups. If the above criterion is fulfilled in an inner iteration the maximum number of inner iterations for the next outer iteration is decreased to this number.

The calculation of the A-coefficients is not needed in every iteration, if the cross sections and diffusion coefficients are constant. With second-order local flux expansion they have to be calculated only once and with higher-order expansions from the viewpoint of minimizing the computer time an optimal procedure would be to calculate them for every third inner iteration. If the calculation of the A-coefficients becomes less frequent the number of outer iterations drastically increases and the convergence is uncertain.

In the outer iterations the eigenvalue λ is calculated. The number of outer iterations is limited through the input, but the iterations also stop when

$$| \lambda(s) - \lambda(s-1) | < dL \qquad (4.6)$$

is fulfilled at the same time as the convergence criterion for the fluxes is fulfilled. s is the outer iteration number. For

every five outer iterations, a new eigenvalue is calculated by an Aitken extrapolation:

$$\lambda(s) = \frac{\lambda(s-2) \cdot \lambda(s) - \lambda(s-1) \cdot \lambda(s-1)}{\lambda(s-2) + \lambda(s) - 2 \cdot \lambda(s-1)} \qquad (4.7)$$

where $\lambda(s-2)$, $\lambda(s-1)$ and $\lambda(s)$ are three successive calculated eigenvalues. After an extrapolation of the eigenvalue is completed a larger number of inner iterations are made to obtain the full benefit of the better eigenvalue. The number is proportional to the third root of the total number of nodes and is at least 20.

4.3. Steady-state input and initialization

All inputs except the array dimensions as described in Section 4.1 are read in the subroutine START, described in Appendix D. Here some of the parameters will be described further. All input variables are printed as soon as they are read.

It is possible to make a dump of all variables after a certain amount of computer time. The time for dump, TDUMP has to be less than the overall computer time limit, because the dump is made after a full outer iteration. By restart nearly all variables are read from dump-file and this is done in the subroutine RESTAR.

The geometric shape of the physical problem can be widely different. It is possible to take account of different symmetry properties by the input variables ISYM and NSYM. The most common types of symmetry properties and their influence on the input variables are shown in Fig. 4.5. It is important to take symmetry properties of a problem into account, because it greatly reduces computer space and time.

The boundary conditions for a case is given by some albedo-matrices and it is possible to have different albedo-matrices

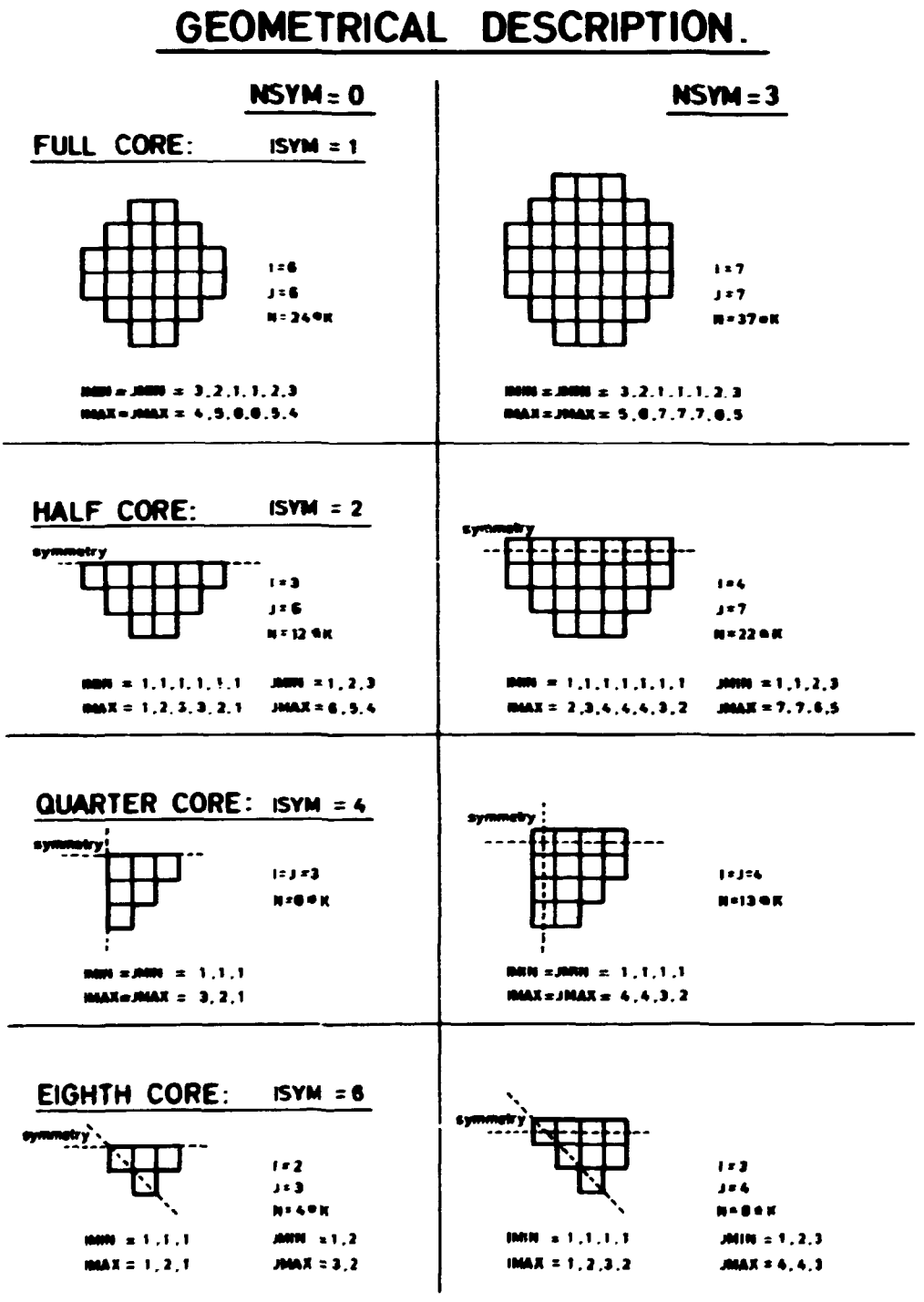


Fig. 4.5. Some of the most common possibilities for utilization of symmetry properties for a certain problem.

in all boundary-channels (no z-dependence is possible). Most problems are in two energy groups and some special cases are described below

$$\text{Albedo-matrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \tag{4.8}$$

1. Fully reflecting boundaries: $J^{in} = J^{out}$

$$\alpha_{11} = \alpha_{22} = 1.0$$
$$\alpha_{12} = \alpha_{21} = 0.0 \tag{4.9}$$

2. No incoming currents: $J^{in} = 0$

$$\alpha_{11} = \alpha_{12} = \alpha_{21} = \alpha_{22} = 0.0 \tag{4.10}$$

3. No flux at the boundaries: $\phi_b = 0$

$$\alpha_{11} = \alpha_{22} = -1.0$$
$$\alpha_{12} = \alpha_{21} = 0.0 \tag{4.11}$$

4. The boundary condition given by a gamma-matrix, i.e. as input to the TWODIM-program (Lindstrøm Nielsen, 1971).

   The gamma-matrix is defined by

$$J = \Gamma \cdot \phi$$

or

$$\begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \begin{pmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{pmatrix} \cdot \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \tag{4.12}$$

Normally there is no upscattering in the reflector, so $\gamma_{12} = 0$. With this assumption the albedo-matrix can be calculated by use of the equations from the diffusion theory

$$\alpha_{11} = \frac{1 - 2 \cdot \gamma_{11}}{1 + 2 \cdot \gamma_{11}}$$

$$\alpha_{12} = 0 \qquad (4.13)$$

$$\alpha_{21} = \frac{-4 \cdot \gamma_{21}}{(1 + 2\gamma_{11})(1 + 2\gamma_{22})}$$

$$\alpha_{22} = \frac{1 - 2 \cdot \gamma_{22}}{1 + 2 \cdot \gamma_{22}}$$

For two-dimensional problems it is possible to give a buckling instead of an albedo for the z-direction.

Besides the description of the problem some variables to be used in the iterations are given by input (see the input description in Appendix D). At last a number of options for the amount of output have to be added. In one end it is possible to print nearly all variables at every inner iteration (a huge amount of output) and at the other end only the input is printed.

The flux-distribution is initialized to unity in the fast group (Group 1). The other groups are initialized by

$$\phi_g^m = \frac{\Sigma_{g1}^m}{\Sigma_{ag}^m} \cdot \phi_1^m \qquad (4.14)$$

The outgoing currents are initialized to

$$J_g^{out,m} = \frac{1}{4} \cdot \phi_g^m \qquad (4.15)$$

For 2 groups both are in agreement with the diffusion theory.

The array WEIGHT are initialized in accordance with the symmetry properties for the problems. For nodes with one symmetry line through the node center WEIGHT = $\frac{1}{2}$ and for nodes with two symmetry lines through the center WEIGHT = $\frac{1}{4}$ or $\frac{1}{8}$ for $\frac{1}{8}$ -case.

4.4. The calculation of incoming currents

The incoming currents are calculated from outgoing currents in the subroutine INCURR. For inner boundaries there are no problems - the incoming currents are the outgoing currents of adjacent nodes relaxed in accordance with Eq. 4.4. At the boundaries of the case the incoming currents are calculated by use of an albedo-matrix and the result is still relaxed as described in Eq. 4.4.

In problems with symmetry, this symmetry is easily utilized by the incoming and outgoing current notation. The incoming current of a node at a symmetry line is simply the outgoing current of its symmetrical counterpart.

For two-dimensional problems with a buckling, (El-Wakil 1962, p. 109) the buckling is converted into an albedo-matrix in the z-direction.

The basic z-dependent solution is

$$\phi(z) = C \cdot \cos(B_z \cdot z)$$

The partial currents are obtained from diffusion theory (Glasstone and Edlund, 1961, p. 111)

$$J_{out} = \frac{\phi}{4} - \frac{D}{2} \cdot \frac{d\phi}{dz}$$

$$J_{in} = \frac{\phi}{4} + \frac{D}{2} \cdot \frac{d\phi}{dz}$$

$$\text{Albedo} = \frac{J^{in}}{J^{out}} = \frac{\Phi + 2 \cdot D \cdot \frac{d\Phi}{dz}}{\Phi - 2 \cdot D \cdot \frac{d\Phi}{dz}}$$

$$= \frac{C \cos(B \cdot z) - 2 \cdot D \cdot C \cdot B \cdot \sin(B \cdot z)}{C \cos(B \cdot z) + 2 \cdot D \cdot C \cdot B \cdot \mathrm{Sin}(B \cdot z)}$$

$$= \frac{1 - 2 \cdot B \cdot D \cdot \tan(B \cdot z)}{1 + 2 \cdot B \cdot D \cdot \tan(B \cdot z)}$$

The albedo has to be calculated at the node boundaries, so $z = \frac{1}{2} \cdot DZ$, where DZ is the node length in the z-direction, which most suitably is set equal to the node length in the x-direction and/or y-direction.

4.5. Generation of A-coefficients

The calculation of the A-coefficients is done in the subroutine A012. A route diagram of the routine is shown in Fig. 4.6.

The expansion coefficients for the transverse leakage approximation (Eq. 3.23) are calculated in the subroutine TRANS by use of Eq. (3.25). The transverse leakage at the boundaries is set to half the transverse leakage of the outermost node.

The higher-order A-coefficients A_3 and A_4 are calculated in a special subroutine A34. An equation system of order G has to be solved for each node and each coordinate. To solve the equation system,

$$A \cdot x = B$$

the square matrix $\underline{A}$ is $\underline{L} \cdot \underline{U}$ factorized by Gaussian elimination with partial pivoting in the subroutine SOLVE. In the special case with only two energy groups the solution is directly calculated (the solution for two equations with two unknowns,.

Coefficients (A012)

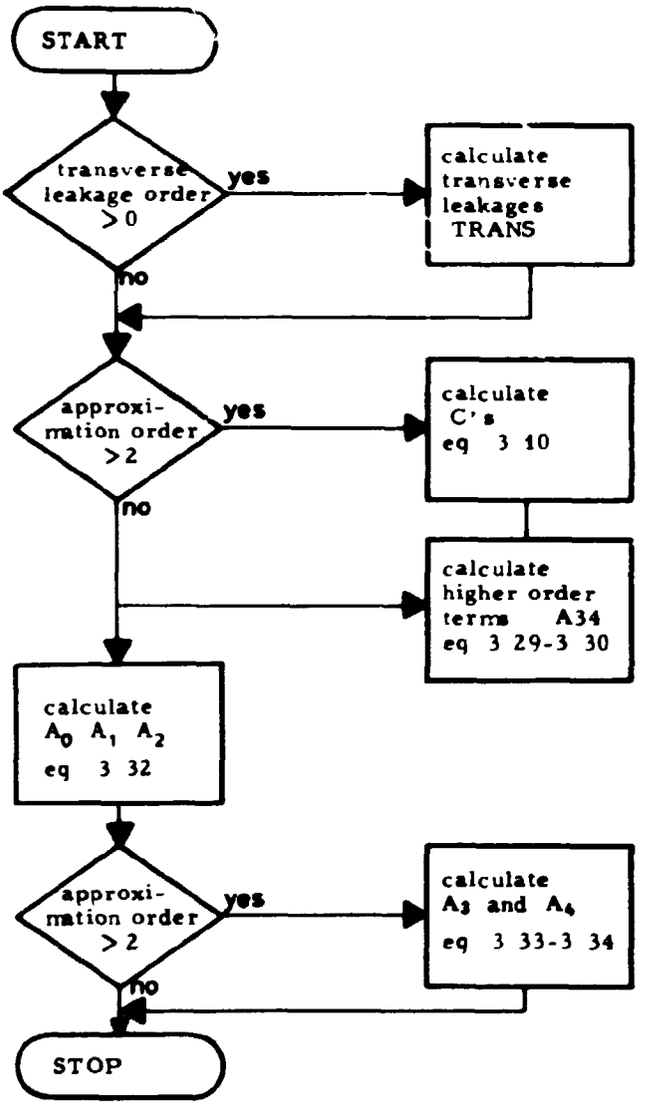


Fig. 4.6. Flow-chart for the subroutine A012, which calculates the A-coefficients.

4.6. Inner iterations

One inner iteration is made in the subroutine FLUXIT. For each node the equation system (3.35) is arranged and solved by the same routine as used in Section 4.5. Then the expression for the outgoing currents are calculated and the convergence criterion for the fluxes (4.5) is controlled. At last the A-constant (4.3) is calculated.

4.7. Refinement

After a fully converged solution to a steady-state problem it is possible to calculate a flux-distribution with finer nodes. The number of nodes in each coordinate can be doublet, triplet, etc.; the nodes are independent of one another. The generation of new input data for the new nodes from the old ones are done in the subroutine PART. The start distributions for fluxes, currents etc. are the solutions from the coarser mesh calculation. A special case is taken to treat symmetry properties, e.g. if the old problem was a quarter case with symmetry lines in node centers, the new case will have symmetry lines at node edges if all nodes are halved in each direction.

The new total number of nodes still has to observe the absolute limit for FORTRAN-arrays as described in Appendix D, so it could be impossible to make refinements with full case problems. For smaller problems, e.g. quarter cases, a refinement with a factor two in all directions is possible, and it gives a good opportunity to check the results. On the other hand, a factor two in all directions provides about 8 times as many unknowns and equations to be solved, which greatly increases the computer consumption, so it is usually not practicable to continue calculations with finer nodes.

4.8. Dynamic main program

The dynamic main program for the nodal expansion method is also the steering routine for the dynamic calculations. The dynamic program is an extension of the steady-state program, so the arrays that are described in 4.1 are also used here. In addition some other global arrays are declared, which mainly contain fluxes and currents from preceding time-steps, and information about delayed neutrons (see Table 4.2). The meaning of the letters in Table 4.2 are

G number of energy groups
I number of nodes in the x-direction
J number of nodes in the y-direction
K number of nodes in the z-direction
L number of delayed neutron families

Table 4.2

List of special global arrays in the dynamic program

Name	Dimensions	Content
FIO	G,K,J,I	Average fluxes in each node and energy group one step back.
FI1	G,K,J,I	Averages fluxes two steps back.
CXL0	G,K,J,I	Average partial currents out of each node in the x-direction at the left edge one step back.
CXL1	G,K,J,I	Same, but two steps back.
CXR0	G,K,J,I	Average partial currents out of each node in the x-direction at the right edge one step back.
CXR1	G,K,J,I	Same, but two steps back.

Name	Dimensions	Content
CYLO	G,K,J,I	Average partial currents out of each node in the y-direction at the left edge one step back.
CYL1	G,K,J,I	Same, but two steps back.
CYRO	G,K,J,I	Average partial currents out of each node in the y-direction at the right edge one step back.
CYR1	G,K,J,I	Same, but two steps back.
CZLO	G,K,J,I	Average partial currents out of each node in the z-direction at the left edge one step back.
CZL1	G,K,J,I	Same, but two steps back.
CZRO	G,K,J,I	Average partial currents out of each node in the z-direction at the right edge one step back.
CZR1	G,K,J,I	Same, but two steps back.
DFIDT	G,K,J,I	The approximation to the time-differentiated flux divided by the average neutron velocity.
C	L,K,J,I	Average precursor distribution in each node and family.
CO	L,K,J,I	Same, but one step back.
BETA	L	The fraction of delayed neutrons in each family.
DECAY	L	The decay constants for the delayed neutrons.
EXPDEC	L	exp(-DECAY*DT).
V	G	The average neutron velocity for each energy group.

Before a dynamic calculation can be performed a successful steady-state calculation has to be made with a dump of the final solution. The dynamic calculation then starts from the

steady-state solution with some additional input (see the input description for dynamic calculations).

A route diagram for the dynamic main program is shown in Fig. 4.7. The calculations of cross sections and diffusion coefficients and their variation with time are made in the subroutine PARAM. This subroutine must be made specially for each problem. No streamlined input routine has been made because various hydraulic feedbacks will have to be added when the routine is built together with a hydraulic reactor model.

In the subroutine PREC the delayed neutron destribution is calculated from the preceding step. The initial precursor distribution is assumed to be in equilibrium.

The calculation of the A-coefficients is made in the subroutine DACOFF, which is much like A012 in the steady-state calculations. It has to include only contributions from delayed neutrons and from the time-differentiated flux. In dynamic calculations the A-coefficients have to be re-calculated for each inner iteration to guarantee convergence.

When the A-coefficients are calculated a new flux-solution and new outgoing currents are calculated in the subroutine INTEGR. The flux convergence is also checked in INTEGR, and there are two different convergence criterions, which can be selected.

If MODE = 0 the convergence criterion is

$$\frac{\sum_{n,g} \left| \phi^{(r)}_{n,g} / \phi^{(r-1)}_{n,g} - 1 \right|}{N \cdot G} < DF \qquad (4.16)$$

where the summation is over all nodes and groups.

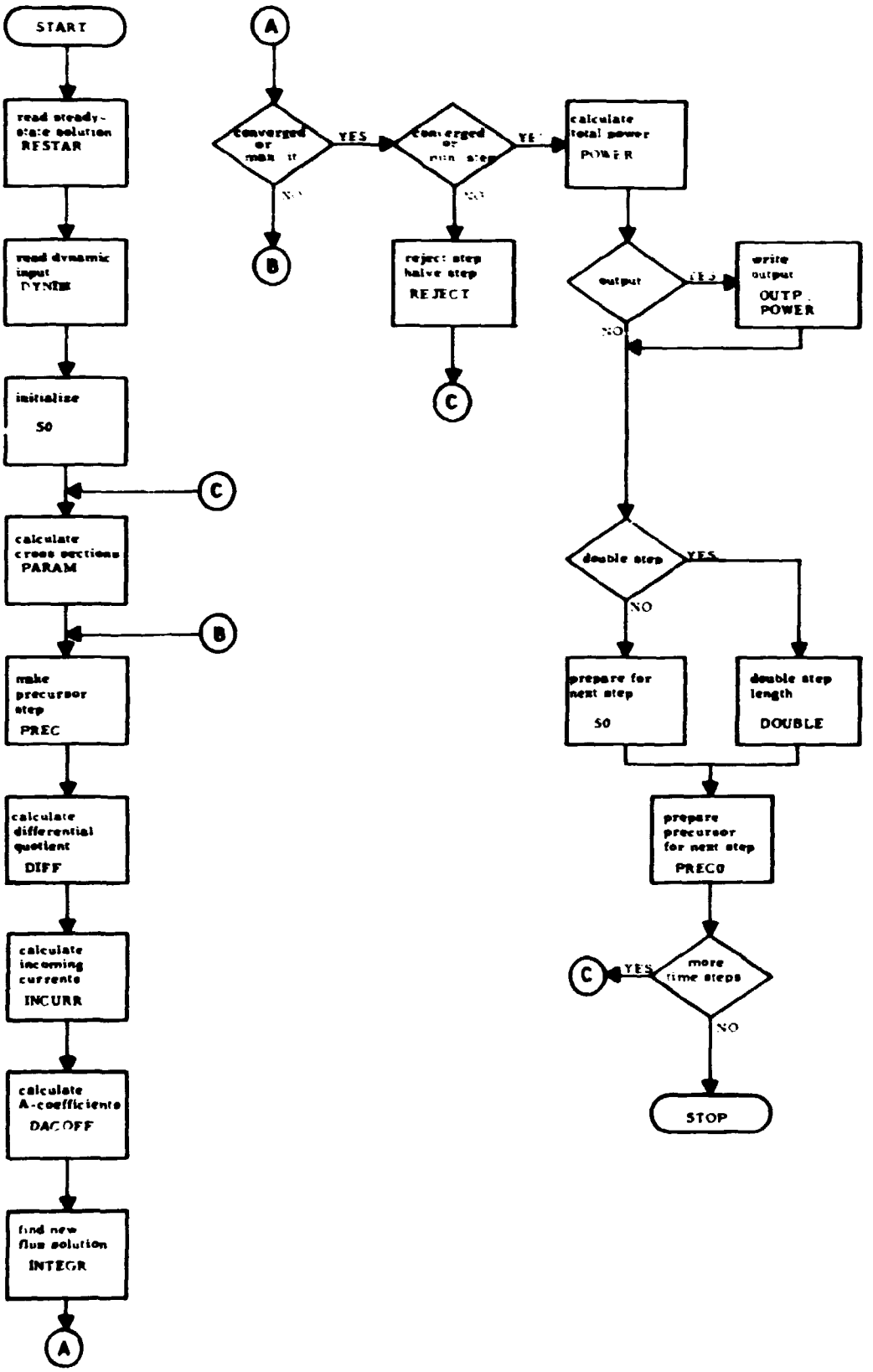


Fig. 4.7. Flow-chart for the main program for dynamic calculations.

If MODE = 2 the convergence criterion is

$$| \phi(r) - \phi(r-1) | < dP \qquad (4.17)$$

for all nodes and energy groups. r is the inner iteration number. Criterion 4.16 is less stringent than 4.17 for the same tolerance, DF.

The inner iterations are continued until the convergence criterion is fulfilled or the maximum number of inner iterations is reached, as specified in the dynamic input.

If the fluxes have not converged after the maximum number of inner iterations, the step is rejected and the step length is halved. However, the step length cannot be halved indefinitely, so a minimum step length can be specified in the input.

When it is possible to decrease the time step length, it also must be possible to increase it again. The step length is doubled when a number of satisfactory time steps have been taken (the number is specified in the input) and when the number of inner iterations per time step is less than one-third of the maximum number. The last condition is given to prevent cycling of the program between two step lengths with many rejected steps as result.

After a successful step all variables are moved to get ready for the next step.

To save inner iterations the fluxes and outgoing currents are linearly extrapolated as a form of predictor step

$$\phi(t+\Delta t_2) = (1 + \frac{\Delta t_2}{\Delta t_1}) \cdot \phi(t) - \frac{\Delta t_2}{\Delta t_1} \phi(t-\Delta t_1)$$

where Δt_1 is the preceding step length and Δt_2 is the new step length.

For each time step the step length and the total power are printed, but larger amounts of output e.g. power and/or flux distributions can be printed with regular intervals. The intervals are specified in the input.

5. EXAMPLES

In this chapter the results of the calculations with the NEM-program will be reported. Some of the results are obtained with special versions of the program to enable special effects to be shown.

The program has been tested on various benchmarks which all are described in Appendix B. Reference solutions and calculated solutions are shown in Appendix C.

The solutions are usually given with the eigenvalue (k_{eff}) and the power distribution and their deviations from the reference solution.

eigenvalue: λ

eigenvalue error: $\Delta\lambda = \lambda - \lambda_{ref}$ normally given in pcm (10^{-5})

power distribution: P

maximum relative error:

$$\Delta P_{max} = \max_{m} \frac{|P_n - P_{ref,n}|}{P_{ref,n}}$$

where n is all nodes. ΔP_{max} is normally given in per cent.

Average relative error:

$$\Delta P_{av} = \frac{1}{N} \cdot \sum_{n=1}^{N} \frac{|P_n - P_{ref,n}|}{P_{ref,n}}$$

also given in per cent.

Two-norm of relative error:

$$\Delta P_2 = \sqrt{\sum_{n=1}^{N} \left(\frac{P_n - P_{ref,n}}{P_{ref,n}}\right)^2}$$

For three-dimensional benchmarks the power distribution is given only for assemblies (average over vertical channels).

All CPU-times are on a Burroughs B7800 computer.

5.1. The IAEA-2D benchmark

The IAEA benchmark is one of the most famous (Micheelsen, and Neltrup, 1973). The two-dimensional benchmark is the midplane of the three-dimensional one. The benchmark is described in Appendix B.1 and a reference solution in Appendix C.1. The benchmark represents a reactor with a two-zone core and a water reflector. Nine fully inserted control rods are represented as smeared absorbers in a single node. Because of the control rods and the water reflector severe flux tilts occur which make the problem quite challenging.

A picture of the power distribution for one quarter of the core is shown in Fig. 5.1. The figure gives a good impression of the differences in the power level in the nodes. The flux depressions in the four nodes with control rods are clear.

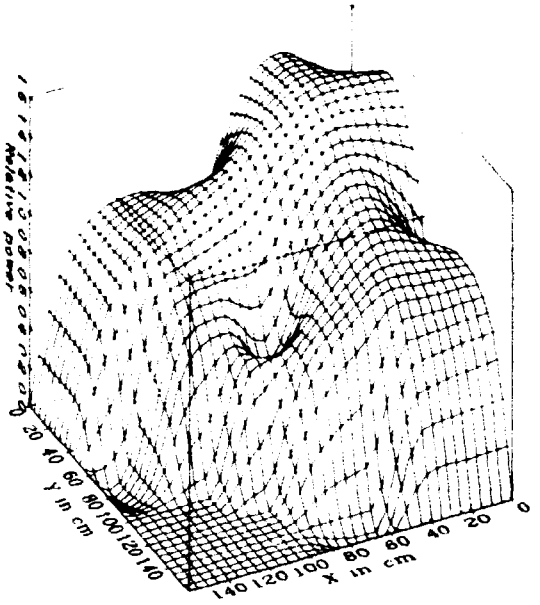


Fig. 5.1. The power distribution for one quarter of the core for the two-dimensional IAEA benchmark. The distribution is normalized to 1.0 for the active part of the core, which is the part that has nodes with power greater than zero ($\Sigma_f > 0$).

5.1.1. Orders and convergence

The IAEA-2D benchmark has been solved with different orders and different node sizes. A summary of the results are shown in Table 5.1, where the first number in the order column is the local flux expansion order and the second number is the transverse leakage expansion order. The transverse leakage expansion order is at least two less than the local flux expansion order, because there is no advantage, for example, in using second-order transverse leakage expansion together with third-order local flux expansion. The extra coefficient is simply unnecessary.

For each order the problem has been solved with different node-sizes by use of the refinement feature in the program. When +24 outer iterations are stated in the table it means that additional iterations are made and for the CPU (Computer Processing Units) as well.

The errors in the eigenvalue as a function of the node size for the various orders are drawn in Fig. 5.2, and in Fig. 5.3 the maximum relative error in the power distribution. The errors here mean the differences from the reference solution (Appendix C.1). There could be objections to making a comparison with a solution that is obtained with another nodal program, but the chosen reference solution is in good agreement with those obtained by quite different methods (e.g. finite-difference methods).

From Fig. 5.2 and 5.3 it is seen in the first place that the errors descend for smaller and smaller node sizes. In other words the solution converges towards the correct solution, when the node size moves toward zero. It is an essential characteristic of a good method.

Secondly, the errors descend when higher-order expansions are used. With coarse meshes we have to use fourth-order local flux expansion and first- or second-order transverse leakage expansion to obtain satisfactory results. Unfortunately, it is two to three times more expensive to use fourth-order rather than second-order expansions, because of the higher-order expansion coefficient calculations. With second-order and constant cross sections we need only to calculate the coefficients once.

In Appendix C.1 a typical solution together with the absolute error is shown. The errors are largest in the nodes facing the reflector. The error distribution is shown in Fig. 5.4.

Table 5.1. Solutions to the IAEA-2D benchmark for various orders and node sizes.

11 inner iterations pr outer

2 inner iterations pr coefficient updating (except for second order)

MODE = 2

Tolerance on eigenvalue: DL = 10^{-6}

Tolerance on flux: DF = 10^{-4}

Orders	Node size cm	Outer iterations	λ	Δλ ·10^5	ΔP_{max} %	ΔP_2 %	CPU sec
2,0	20.0	32	1.02873	-86	22.9	55.8	14.4
	10.0	+24	1.02901	-58	16.9	42.9	+26.4
	5.0	+39	1.02943	-16	7.7	20.8	+168.2
	2.5	+40	1.02966	7	3.5	11.1	+684.6
3,0	20.0	43	1.02953	-6	13.0	29.8	27.5
	10.0	+15	1.02974	15	4.6	12.3	+26.7
	5.0	+34	1.02969	10	1.7	5.2	+182.3
	2.5	+35	1.02966	7	0.97	3.2	+701.8
3,1	20.0	47	1.02913	-46	13.3	32.3	33.9
	10.0	+22	1.02959	0	3.4	9.9	+50.6
	5.0	+15	1.02970	11	1.7	5.9	99.5
	2.5	+40	1.02966	7	1.1	3.6	+946.0
4,0	20.0	42	1.02996	37	2.6	5.6	36.4
	10.0	+9	1.02973	14	1.2	2.5	+19.9
	5.0	+7	1.02963	4	0.56	1.1	+82.4
	2.5	+34	1.02960	1	0.20	0.34	+941.7
4,1	20.0	47	1.02955	-4	1.6	4.1	44.5
	10.0	+5	1.02968	9	0.97	2.8	+15.5
	5.0	+19	1.02962	3	0.46	1.6	+59.3
	2.5	+30	1.02961	2	0.31	1.0	+941.9
4,2	20.0	45	1.02949	-10	1.3	2.8	45.8
	10.0	+5	1.02966	7	0.77	1.6	+16.7
	5.0	+14	1.02960	1	0.29	0.75	+134.9
	2.5	+29	1.02959	0	0.13	0.39	+979.0

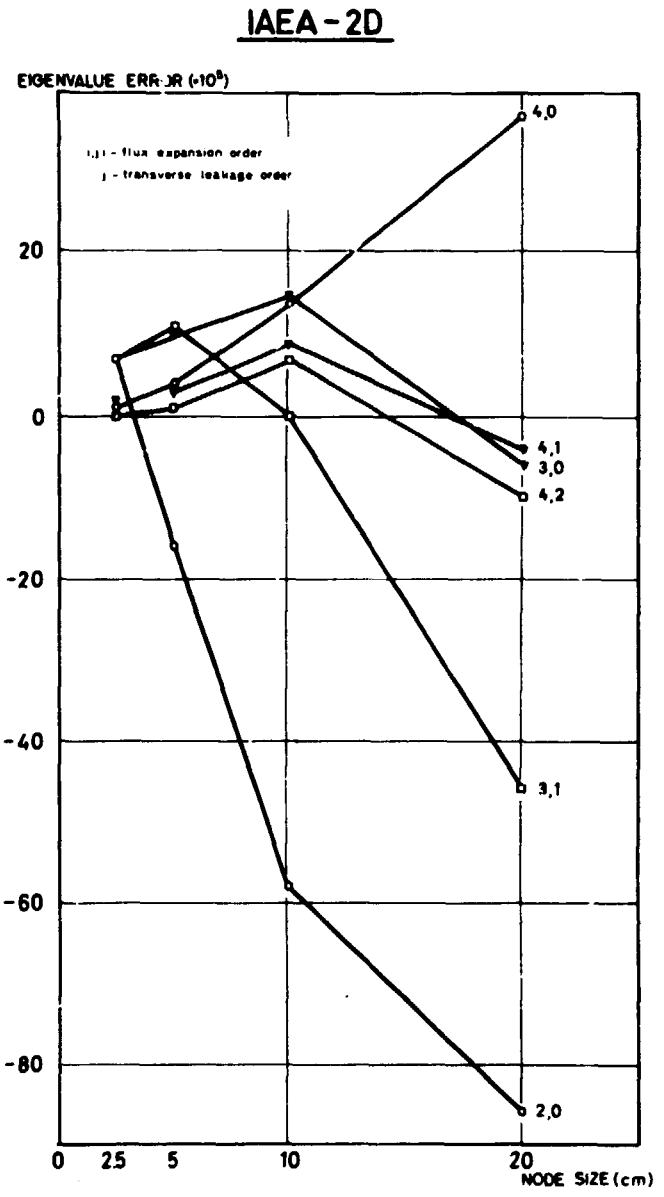


Fig. 5.2. The errors in the eigenvalue as function of the node size for various orders for the IAEA-2D benchmark.

5.1.2. Mode, tolerance and transverse leakage

There are several parameters in the iteration process that can be changed in different ways. Unfortunately, they are not independent, so when one parameter is optimized the others may not have retained their optimal values.

MODE is the parameter which determines how the inner iterations should be normalized, and if they should be accelerated. The accelerated iterations could be used only with second-order

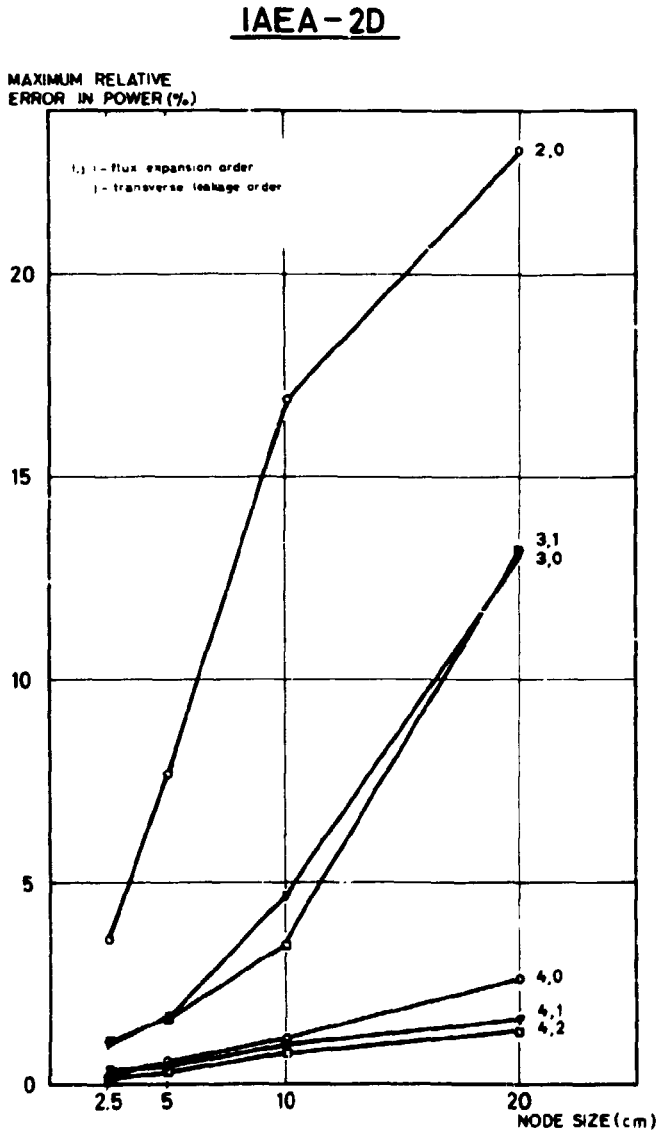


Fig. 5.3. The maximum relative error in the power distribution as function of the node size for various orders for the IAEA-2D benchmark.

local flux expansions. With higher orders the iterations will not converge. In Table 5.2 results with second-order local flux expansion and different values of MODE are shown. The constant γ is 0.8, which is near the optimum value for this example. The accelerated method is a little quicker, and it is also quicker to use the maximum flux to normalize upon (MODE = 2 and 3) than the average.

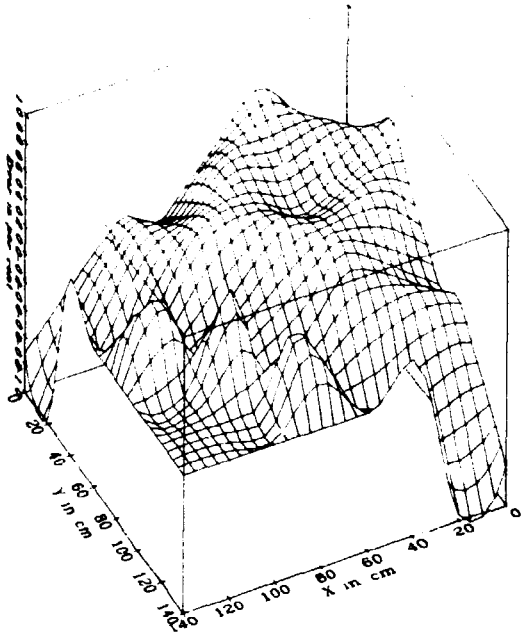


Fig. 5.4. The absolute error distribution for one quarter of the core for the two-dimensional IAEA benchmark.

With higher orders especially fourth order we have only to consider MODE = 0 or MODE = 2. In Table 5.3 the IAEA-2D problem is solved for different values of MODE. Strengthening the tolerance requirement is also tried, but this has no effect on the accuracy of the solution, and it takes longer time.

The column LEAK illustrates two different ways of treating the transverse leakage expansion at the boundary of the core. LEAK = 1/2 means that the transverse leakage at the outer boundary of the outermost node is set to half the average value. LEAK = 0 means that it is set to zero. There is no significant difference and the standard is set to LEAK = 1/2.

Nor does it matter which MODE is used. However, MODE = 2 is usually a little faster than MODE = 0.

5.1.3. Better start-values and inner iterations

In Paragraph 5.1.1 it was shown that it was necessary to use fourth-order local flux expansion order and first- or second-order transverse leakage order. Unfortunately, fourth-order ex-

Table 5.2. IAEA-2D benchmark. 20-cm nodes second-order local flux expansion order.
10 inner iterations per outer ones.
Tolerance on eigenvalue = 10^{-6}.
Tolerance on power = 10^{-4}.

Mode	γ	Outer Iterations	λ	Δλ ×10^5	ΔP_{max} %	ΔP_2 %	CPU sec
0	0	35	1.02875	-84	23.1	56.2	15.5
1	0.8	32	1.02873	-86	22.9	55.8	14.4
2	0	34	1.02872	-87	22.9	55.5	15.3
3	0.8	24	1.02862	-97	22.3	52.5	10.4

Table 5.3. IAEA-2D benchmark. 20-cm nodes.
Fourth-order local flux expansion.
Second-order transverse leakage expansion.
10 inner iterations pr outer.

Mode	Tolerance DL	Tolerance DF	Leak	Outer iterations	λ	Δλ ·10^5	ΔP_{max} %	ΔP_2 %	CPU sec
0	10^{-6}	10^{-4}	1/2	47	1.02951	-8	1.35	3.0	47.6
0	10^{-7}	10^{-5}	1/2	82	1.02953	-6	1.52	3.4	79.8
2	10^{-6}	10^{-4}	1/2	45	1.02949	-10	1.30	2.8	45.8
2	10^{-7}	10^{-5}	1/2	63	1.02953	-6	1.33	3.3	64.6
0	10^{-6}	10^{-4}	0	44	1.02950	-9	1.50	3.5	47.9
2	10^{-6}	10^{-4}	0	43	1.02950	-9	1.60	3.3	46.2

pansions are rather expensive, so here an attempt is made to speed-up the calculations by using the second-order method in the beginning to a certain point and then let the fourth-order take over.

In Table 5.4 the results of such a strategy are shown, where $ITOUT_2$ iterations are taken with 2,0 orders method and $ITOUT_4$ iterations with 4,2 orders method. UPDATE is the number of iterations between calculations of the A-coefficients, and it has its optimum at 3. For longer intervals between calculations the iterations would not converge.
The tolerance demand for the 2,0 orders method is lessoned by some factors in proportion to the usual ones.

Tolerance on eigenvalue: $DL_2 = FDL_2 \cdot DL_4$
Tolerance on flux: $DF_2 = FDF_2 \cdot DF_4$

The optimum values are $FDL_2 = 100$ and $FDF_2 = 12$.

At last the number of inner iterations per outer ones can be varied. The total computer time does not vary unambigiously with the number, but there seems to be a minimum point around 20. The optimum value for this example is 17.

These improvements have decreased the computer time by a factor of three without losing accuracy. The optimum parameters found may not be optimum for other problems, but they can still manage quite well.

Table 5.4. IAEA-2D benchmark. 20-cm nodes.
MODE = 3 (γ = 0.8) for second-order and MODE = 2 for fourth-order.

Inner Iteration	Update	FDL_2	FDF_2	$ITOUT_2$	CPU_2	$ITOUT_4$	$\Delta\lambda$ $\cdot 10^5$	ΔP_{max} %	CPU sec
20	1	100	100	12	4.1	18	-9	1.31	42.2
20	2	100	100	12	4.0	13	-11	1.35	24.3
20	3	100	100	12	4.1	17	-11	1.36	21.1
20	4	100	100	12	4.0	diverge			
20	3	10	100	17	5.0	12	-8	1.42	21.9
20	3	1	100	50	11.9	12	-1	2.28	28.8
20	3	100	10	9	5.7	8	-4	1.78	16.9
20	3	100	1	14	10.7	12	-1	2.27	27.7
20	3	100	12	7	5.1	7	-9	1.25	14.4
15	3	100	12	9	5.0	17	-8	1.42	20.3
10	3	100	12	11	5.3	12	-8	1 33	15.5
25	3	100	12	7	5.7	12	-5	1.74	25.6
17	3	100	12	9	5.2	7	-8	1.20	13.4

5.2. The IAEA-3D benchmark

The IAEA-3D benchmark is described in Appendix B.2. The example is much like the two-dimensional one but four partially inserted control rods are also inserted. The benchmark has been calculated with 20 x 20 x 20 cm^3 nodes and the results are collected in Table 5.5.

The results marked with a star are obtained with the procedure described in Section 5.1.3 with the optimum iteration parameters from the 2D-IAEA benchmark. The last solution together with the absolute errors are shown in Appendix C.2. It is seen, that there

are no basic differences between the results for the two-dimensional benchmark and the results for the three-dimensional benchmark, neither of which were expected. Still a saving of about a factor 3 in computer time was obtained by using second order in the first iterations.

Table 5.5. IAEA-3D benchmark. 20-cm nodes.
MODE = 2
Tolerance on eigenvalue: DL = 10^{-6}
Tolerance on flux: DF = 10^{-4}

Orders	Outer iterations	λ	$\Delta\lambda$ $\cdot 10^5$	ΔP_{max} %	ΔP_{av} %	CPU sec
2,0	44	1.02797	-106	22.75	8.45	235
4,2	62	1.02896	-7	1.42	0.77	1235
2,0*	12	1.02685	-218	19.38	5.41	115
4,2*	+17	1.02883	-15	1.35	0.66	+347

5.3. The Biblis 2D-benchmark

The Biblis benchmark is described in Appendix B.3. The Biblis benchmark is in two configurations, one without a control rod and one with one in a node represented by a larger absorption cross section in that node. Moreover, it is possible to represent the reflector in three different ways: with an albedo value, with one zone of reflector cross sections, and with two zones. Illustrations of the power distributions are shown in Figs. 5.5 and 5.6.

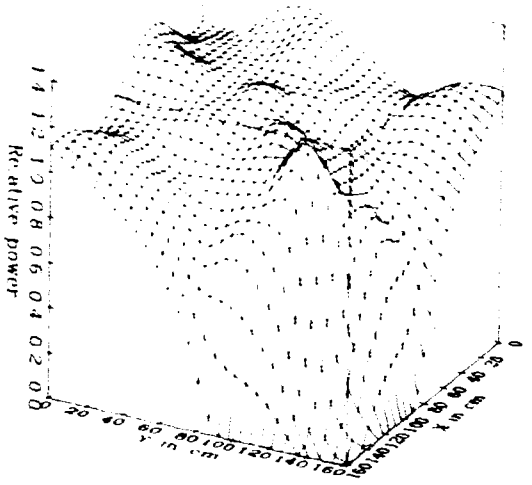


Fig. 5.5. The power distribution for one quarter of the core for the two-dimensional Biblis benchmark. The control rod is out (BIBLIS-1).

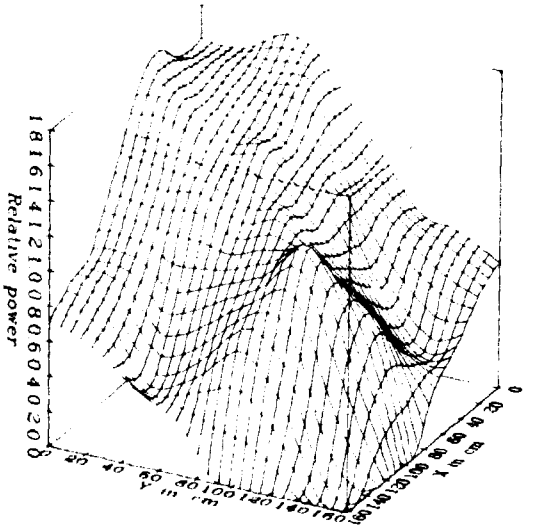


Fig. 5.6. The power distribution for one quarter of the core for the two-dimensional Biblis benchmark. The control rod is in (BIBLIS-2).

The calculated distributions are compared with solutions obtained with the finite-difference program TWODIM (Lindstrøm Jensen, 1971) and TVEDIM (Kristiansen, 1977). The reference solutions are obtained with a mesh size of 2.89 cm. The solution for problems 1 and 3 and respectively 2 and 4 cannot be compared directly when there is no assurance of good correspondence between reflector cross sections and albedos.

Reference solutions:

Problem	Program	K_{eff}
Biblis-1	TVEDIM	1.02398
Biblis-2	TVEDIM	1.01867
Biblis-3/1	TWODIM	1.02520
Biblis-3/2	TVEDIM	1.02506
Biblis-4/1	TWODIM	1.01970
Biblis-4/2	TVEDIM	1.01935

The benchmarks have been solved with orders 4,2 and with normal tolerances (10^{-6} and 10^{-4}) for different node sizes. The results are given in Appendix C.3 and they are summarized in Table 5.6.

The results for the problems with reflector zones (3 and 4) are in good agreement with those obtained with the finitedifference code. If one or two reflector zones are used this has nearly no effect. The two solutions are identical to within a few tenths of one per cent. The deviations in Table 5.6 are merely caused by the different reference solutions. The results are nearly as good as they can be even with coarse meshes.

It is quite another case when the reflector is represented with albedo matrices (1 and 2). Here there are deviations of about 10% in the power distribution for large nodes, and the deviations decrease to about 3% when each node is divided into 16 smaller ones. This rather large effect of node size is quite unfortunate considering the number of nodes, because it means that the use

of albedos for the reflector becomes rather difficult. Of course, the reflector cross section representation can be used, but it is, however, difficult to calculate these cross sections.

At last it has been tried to calculate albedos from the partial currents at the boundary to the reflector in a calculation on Biblis-3/2

$$\alpha_g = \frac{j_g^{in}}{j_g^{out}}$$

The calculated albedos are shown in Table 7 for different node sizes, so the variation inside a large node can be seen. The variation in the calculated albedos is largest in nodes at corners of the core, where the deviation is about 20%, but if the four albedos from the fine calculation are averaged they agree with the coarse-mesh albedos to within a few per cent.

The calculated coarse-mesh albedos are used in the Biblis-1 benchmark and the results are compared with the Biblis-3/2 reference solution. The results are shown under 1-A in Table 5.6, and the deviation from the reference solution is (of course) small for large nodes, but it increases for smaller nodes.

The conclusion is that with the reflector represented by albedos the results are rather sensitive to node sizes. A reflector represented with one (or more) reflector zones with cross sections must be preferred, but it also means more nodes to the problem.

Table 5.6. The Biblis-2D benchmark. Fourth-order local flux expansion. Second-order transverse leakage expansion. 17 inner iteration per outer. MODE = 2. Tolerance on eigenvalue: DL = 10^{-6}. Tolerance on flux: DF = 10^{-4}.

Problem	Node size cm	λ	$\Delta\lambda$ $\cdot 10^5$	ITOUT	ΔP_{max} %	ΔP_{av} %	CPU Sec
1	23.12	1.02467	69	22	8.68	2.99	24
	11.56	1.02424	26	+14	3.67	1.18	+41
	5.78	1.02413	15	+7	2.64	0.85	+92
2	23.12	1.01921	54	28	10.68	3.21	29
	11.56	1.01889	23	+14	4.46	1.32	+41
	5.78	1.01881	14	+6	3.25	0.98	+91
3/1	23.12	1.02504	-16	23	1.08	0.49	30
	11.56	1.02511	-9	+3	1.14	0.49	+14
	5.78	1.02512	-8	+7	1.14	0.42	+125
3/2	23.12	1.02505	-1	27	2.35	0.75	36
	11.56	1.02511	5	+8	1.92	0.94	+33
	5.78	1.02513	7	+8	1.91	0.87	+142
4/1	23.12	1.01945	-25	27	1.22	0.55	37
	11.56	1.01948	-22	+7	0.78	0.40	+29
	5.78	1.01950	-20	+6	0.73	0.38	+122
4/2	23.12	1.01944	9	26	1.88	0.65	41
	11.56	1.01948	13	+7	1.58	0.65	+33
	5.78	1.01950	14	+6	1.41	0.62	+139
1-A	23.12	1.02536	30	21	2.12	0.68	25
	11.56	1.02494	-12	+13	5.43	1.83	+42
	5.78	1.02484	-22	+7	6.07	2.10	+97

Table 5.7. Calculated albedos with the Biblis-3/2 benchmark with different mesh sizes.

Albedos in the y-direction

I	Group	23.12 cm	11.56 cm		5.78 cm			
			1	2	1	2	3	4
1	1	0.460	-	0.459	-	-	0.459	0.459
	2	1.002	-	1.024	-	-	1.029	1.029
2	1	0.460	0.459	0.459	0.459	0.459	0.459	0.459
	2	1.001	1.024	1.023	1.029	1.029	1.029	1.029
3	1	0.458	0.458	0.457	0.458	0.458	0.457	0.456
	2	1.000	1.023	1.023	1.029	1.028	1.028	1.028
4	1	0.459	0.453	0.449	0.455	0.452	0.449	0.451
	2	0.988	1.021	1.014	1.027	1.026	1.024	1.010
5	1	0.508	0.557	0.472	0.604	0.513	0.479	0.464
	2	1.016	1.059	1.030	1.040	1.053	1.038	1.031
6	1	0.443	0.452	0.445	0.455	0.449	0.445	0.448
	2	0.986	1.021	1.013	1.027	1.024	1.022	1.009

Albedos in the x-direction

J	Group	23.12 cm	11.56 cm		5.78 cm			
			1	2	1	2	3	4
7	1	0.537	0.580	0.499	0.620	0.535	0.502	0.493
	2	1.025	1.067	1.038	1.076	1.062	1.048	1.030
8	1	0.584	0.634	0.515	0.688	0.570	0.521	0.505
	2	1.044	1.088	1.045	1.100	1.077	1.057	1.036

5.4 A typical Westinghouse PWR

The fourth example is an attempt to compare the calculated assembly power distribution with measurements from a real reactor. To simulate a real reactor it is necessary to be able to take feedback effects from the hydraulics in consideration. To circumvent this difficulty the problem has been calculated with the ANTI-code (Nielsen and Larsen, 1980) and finally, when the code has converged, the cross sections and diffusion coefficients are printed for all nodes. The ANTI-solution is in rather good agreement with the measurements and are also used as reference (see Appendix C.4).

Thorlaksen, 1981 calculated an average gamma-matrix

$$\Gamma = \begin{pmatrix} 0.1491 & 0 \\ -0.0345 & 0.1624 \end{pmatrix}$$

and with an assumption on the proportion Φ_2/Φ_1, he found a one-group albedo to 0.6. By use of the formulas in Chapter 4.3 an albedo-matrix can be calculated.

$$\alpha_T = \begin{pmatrix} 0.5406 & 0 \\ 0.8020 & 0.5097 \end{pmatrix}$$

If we use the same assumption on the flux proportion the albedo-matrix can be diagonalized

$$\alpha_T^D = \begin{pmatrix} 0.5406 & 0 \\ 0 & 0.8576 \end{pmatrix}$$

Calculations with these parameters have been done, and the results are shown in Appendix C.4 and summarized in Table 5.8. With this example there is little benefit is using second-order, first and fourth-order later instead of using fourth-order local flux expansion throughout the calculation.

Secondly, the deviations from the ANTI-solution are quite large, and they decrease only slightly for smaller nodes. Surprisingly,

Table 5.8. Results for example 4 with use of the reflector parameters calculated by Thorlaksen, (1981).

Albedo	Orders	Node size cm^3	Outer iterations	λ	$\Delta\lambda$ $\cdot 10^5$	ΔP_{max} %	ΔP_{av} %	CPU sec
α_T^D	2,0	$21.5^2 \times 18.3$	12	0.99300	300	9.7	4.2	77
	4,2		+31	0.99400	400	13.7	5.9	+563
α_T^D	4,2	$21.5^2 \times 18.3$	38	0.99399	399	13.7	5.9	675
		$10.75^2 \times 9.15$	+14	0.99298	298	10.8	4.6	+1357
α_T	4,2	$21.5^2 \times 18.3$	38	0.99443	443	15.7	6.8	661
		$10.75^2 \times 9.15$	+14	0.99346	346	12.9	5.3	+1372

the calculation with diagonal albedo matrices has the smallest deviations.

The whole assembly power distribution is printed in Appendix C.4 and it is compared with a calculation with QUANDRY (Smith, 1979) and with ANTI. There are very small deviations (less than 2%) between the NEM-solution and the QUANDRY-solution, but they both differ quite much from the ANTI-solution. The NEM-solution of the power distribution has too high power values near the reflector and a depression in the center nodes, which could be due to reflection properties in the reflector that are too good.

An attempt to produce better reflector parameters from the geometrical description of the reflector has been made. The reflector is rather complex with corners and different materials as shown in Fig. B.4. Two different reflector sets have been calculated, one in the axial direction and another average set for the rest of the reflector.

At first the cross sections for the reflector are calculated with the code CRSIQ (Lauridsen, 1977) and condensed to 2 groups. It also calculates homogenised cross sections for the node size

used. The condensed cross sections are used in the program HECS (Pedersen, 1969) that calculates albedo- and gamma-matrices. The reflector configuration used and the results are shown in Table 5.9.

The new calculated reflector parameters are used in NEM and the problems is solved for both albedo-matrices and reflector cross sections. The resulting assembly distributions are given in Appendix C.4. The two NEM-solutions agree quite well except in some of the nodes facing the reflector, but the agreement with the ANTI-solution has not been any better.

In Table 5.10 the average power for each layer in the core for the four NEM calculations and with the ANTI solution as reference. The boundary conditions in the z-direction are albedo-matrices in all four calculations, also in the run with cross sections as reflector representation. The albedo-matrices in the z-direction are the same as used in x- and y-directions since we have no better data. ANTI uses 0.6 in both top and bottom.

The 4 NEM-solutions do not differ very much from each other, but they are flatter than the ANTI-solution, and in this direction this is in better agreement with the measurements.

To make a final test of NEM a two-dimensional example has been made by picking out the middle layer of the core. This problem has then been solved by NEM and by TWODIM (Lindstrøm Jensen, 1971) and compared. Further on, a very fine modelling of the reflector has been made with homogenizing of only each material by itself (water, stainless steel) and with all details in the core baffle and core barrel. It was also solved by TWODIM, but with a very fine mesh (0.67 cm).

The results are shown in Table 5.11 for reflector cross sections and in Table 5.12 for albedo- and gamma-matrices. With homogeneous reflector cross sections the distributions agree within 0.5%, but not with the very fine calculation which is different by about 15% near the reflector.

Table 5.9. New reflector calculations for example 4 with CRSIQ and HECS.

		Average reflector	Axis reflector
Number of regions		24	24
Number of regions in fuel		5	5
Number of regions in core baffle		3	3
Number of regions in water		14	5
Number of regions in core barrel		2	10
Radius of fuel		10.0 cm	10.0 cm
Radius of core baffle		12.5 cm	12.5 cm
Radius of water		27.5 cm	17.9 cm
Outer radius		31.5 cm	31.5 cm
Diffusion coefficients,			
	group 1	1.32122 cm	1.13855 cm
	group 2	0.291509 cm	0.311097 cm
Absorption cross sections,			
	group 1	0.0015836 cm^{-1}	0.0021617 cm^{-1}
	group 2	0.0348048 cm^{-1}	0.050987 cm^{-1}
Down scattering cross sections			
	1→2	0.027926 cm^{-1}	0.0166602 cm^{-1}
Albedo-matrix	α_{11}	0.589999	0.611338
	α_{21}	0.0426159	0.043182
	α_{22}	0.464227	0.465118
Gamma-matrix	γ_{11}	1.28931	1.20602
	γ_{21}	-0.0283048	-0.0182973
	γ_{22}	0.182954	0.182539

Table 5.10. Vertical average powers for example 4. Layer 1 is at the bottom and layer 20 at the top of the core.

Layer	NEM α_T^D	NEM α_T	NEM new albedo	NEM cross sections	ANTI reference
20	0.220	0.226	0.216	0.216	0.192
19	0.428	0.433	0.418	0.421	0.395
18	0.606	0.611	0.593	0.598	0.572
17	0.754	0.759	0.740	0.747	0.725
16	0.878	0.883	0.870	0.878	0.858
15	0.984	0.988	0.979	0.987	0.975
14	1.076	1.079	1.073	1.081	1.079
13	1.156	1.158	1.155	1.163	1.172
12	1.226	1.227	1.226	1.233	1.253
11	1.287	1.286	1.287	1.292	1.322
10	1.336	1.334	1.337	1.341	1.378
9	1.373	1.369	1.375	1.376	1.416
8	1.393	1.388	1.395	1.395	1.433
7	1.390	1.384	1.394	1.391	1.423
6	1.357	1.351	1.363	1.357	1.378
5	1.284	1.278	1.293	1.284	1.291
4	1.161	1.156	1.172	1.161	1.151
3	0.977	0.973	0.990	0.977	0.952
2	0.723	0.721	0.736	0.722	0.686
1	0.393	0.395	0.401	0.382	0.352

Table 5.11. The power distribution for the central layer (10) of example 4 calculated with reflector cross sections.

0.950	0.928	1.021	1.009	1.132	1.210	1.137	0.985
0.951	0.927	1.021	1.005	1.127	1.203	1.128	0.981
0.956	0.927	1.025	1.004	1.131	1.203	1.131	0.980
1.098	1.062	1.152	1.103	1.192	1.211	1.070	0.846
	1.001	1.030	1.076	1.100	1.142	1.101	0.781
	1.001	1.027	1.073	1.094	1.136	1.097	0.787
	1.006	1.028	1.077	1.094	1.140	1.095	0.785
	1.143	1.152	1.175	1.152	1.139	1.022	0.637
		1.070	1.063	1.073	1.013	1.011	
		1.068	1.058	1.070	1.010	1.020	
		1.073	1.058	1.074	1.010	1.019	
		1.183	1.142	1.113	0.993	0.888	
			1.054	0.976	0.957	0.706	
			1.051	0.974	0.961	0.722	
			1.054	0.974	0.960	0.718	
			1.109	0.985	0.906	0.585	

NEM	21.5 cm nodes	0.895	0.757
NEM	5.375 cm nodes	0.894	0.774
TWODIM	1.34 cm	0.895	0.768
TWODIM, fine reference	0.67 cm	0.858	0.640

The eigenvalues are	
NEM coarse mesh	0.99728
NEM fine mesh	0.99754
TWODIM	0.99758
TWODIM, reference	0.99468

Table 5.12. The power distribution for the central layer (2D) of example 4 calculated with albedo- or gamma-matrices as reflector representation.

0.991	0.967	1.060	1.040	1.156	1.225	1.146	1.001
1.039	1.010	1.103	1.071	1.178	1.227	1.122	0.957
1.062	1.026	1.121	1.076	1.182	1.220	1.114	0.946
1.098	1.062	1.152	1.103	1.192	1.211	1.070	0.846
	1.041	1.067	1.106	1.122	1.153	1.095	0.731
	1.088	1.106	1.139	1.138	1.151	1.064	0.707
	1.108	1.118	1.150	1.137	1.148	1.052	0.699
	1.143	1.152	1.175	1.152	1.139	1.022	0.637
		1.103	1.087	1.087	1.017	0.959	
		1.140	1.110	1.096	1.003	0.932	
		1.153	1.114	1.099	0.997	0.926	
		1.183	1.142	1.113	0.993	0.888	
			1.069	0.978	0.944	0.646	
			1.079	0.969	0.910	0.628	
			1.084	0.965	0.904	0.623	
			1.109	0.985	0.906	0.585	

NEM 21.5 cm nodes		0.881	0.666
NEM 5.375 cm nodes		0.840	0.635
TWODIM 1.34 cm		0.836	0.630
TWODIM, fine reference	0.67 cm	0.858	0.640

The eigenvalues are	
NEM coarse mesh	0.99664
NEM fine mesh	0.99539
TWODIM	0.99550
TWODIM, reference	0.99468

With albedo- and gamma-matrices the differences are not quite as large, but in return results differ for different node sizes.

In Fig. 5.7 the two-group fluxes near and in the reflector are drawn when they are respectively calculated with the reflector homogenization code CRSIQ and with TWODIM with the fine reflector modelling. The fluxes are normalized so the group 1 fluxes coincide at the core boundary. The shape of the curves are the same, but the levels in the reflector are not quite the same.

Figure 5.8 shows a picture of the fluxes in a cross section of the whole core calculated with TWODIM. It is seen that the model with the homogenized reflector is unable to predict the local maximum in the thermal flux caused by the water in the reflector. It seems a little thing, but it has a rather large effect of the flux-levels in the middle of the core.

The attempt to compare calculations with real measurements failed, but the code cannot be blamed. The problem is how to treat the reflector and especially how to homogenize mixtures of water and steel. The ANTI-solution agrees with the measurements but ANTI also has several parameters, which can be fitted to the problem.

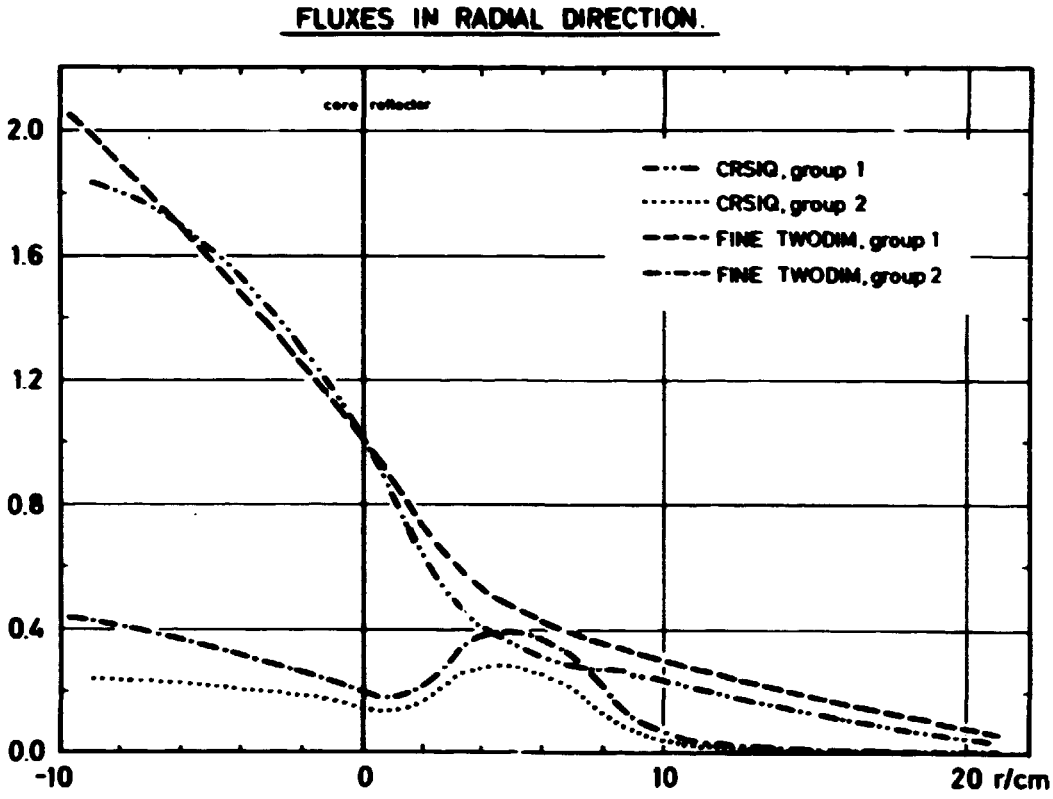


Fig. 5.7. The two-group fluxes in the radial direction around the boundary for the typical Westinghouse PWR. The fluxes are calculated with, respectively, the TWODIM-program and the CRSIQ-program.

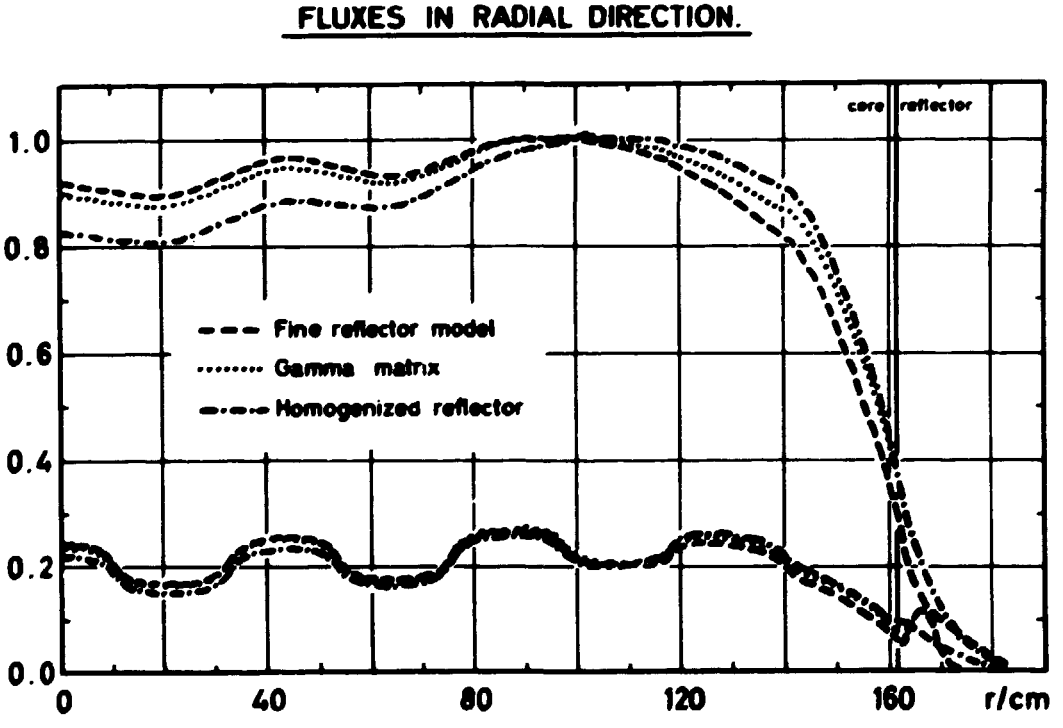


Fig. 5.8. The two-group fluxes in the radial direction for the typical Westinghouse PWR calculated with the TWODIM-program with various reflector strategies.

5.5. Infinite slab reactor model

This very primitive benchmark is from The Benchmark Problem Book (ANL, 1977) and the data for the example are given in Appendix B.5. The example is calculated with 12 nodes each of 20 cm. Whether the slab is in the x, y or z-direction does not matter, since they are symmetric (for the use in this example).

The example is calculated with orders 4,2 and the dynamic solution is given by the relative power fractions in each region. The steady-state solution is

	NEM	Reference
k_{eff}	0.90163	0.90155
Power in region 1	0.278	0.279
Power in region 2	0.443	0.442
Power in region 3	0.278	0.279

The reference solution is obtained with a finite-difference code WIGLE (Cadwell, Henry, Vigibotti, 1964) with 120 2-cm meshes, and the solution is reported in The Benchmark Problem Book (ANL, 1977).

Transient 1, a linear increment of 3% in the thermal absorption cross section in region 1 in 1.0 second, is calculated with a variable step length. The power distribution is printed in Appendix C.5 and a diagram of the total power is shown in Fig. 5.9. The total power decreases more than the reference solution, but it can be due to the slightly larger initial eigenvalue.

In transient 2 the thermal absorption cross section in region 1 is linearly decreased by 1% in 1 second. The power distribution (see Appendix C.5) is also lower than the reference solution in

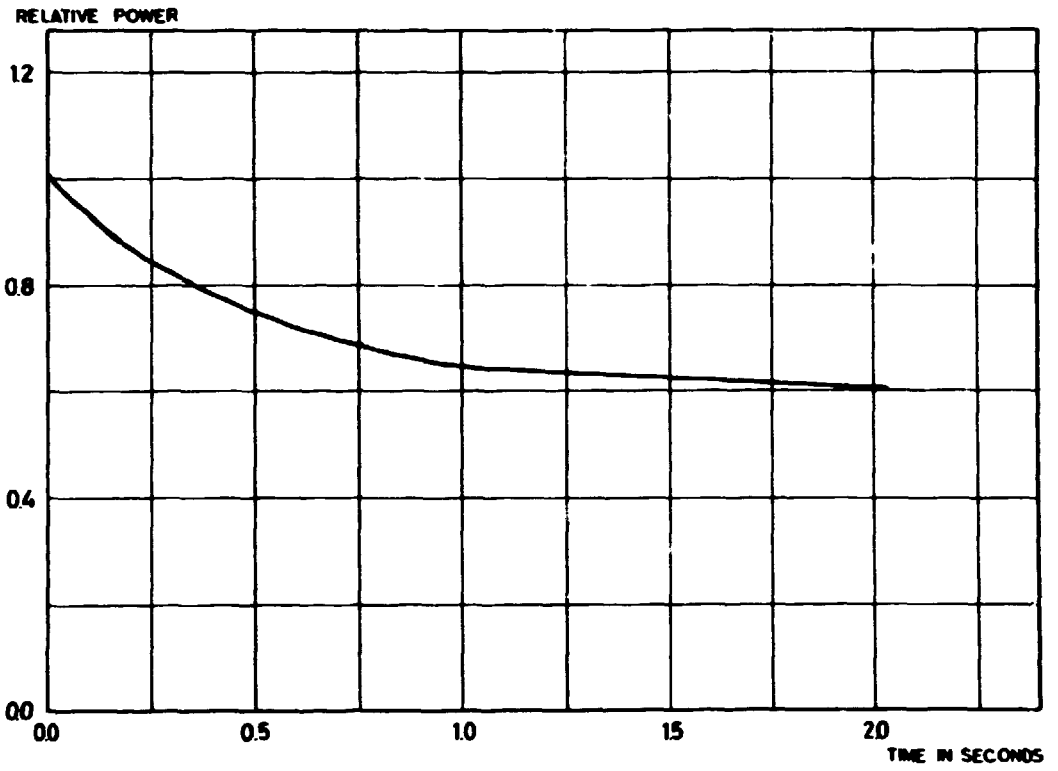


Fig. 5.9. The average power versus time for the slab reactor model, when the thermal absorption cross section in region 1 is linearly increased by 3% in 1.0 second.

this case. In Fig. 5.10 the total power is shown, and it is seen that there is a small numerical overshoot at the end of the cross section change, but the oscillations quickly die away.

The third transient is a prompt supercritical transient, which results in an exponential increase in the power level as seen in Fig. 5.11. The total power corresponds with the reference solution within 1% (Appendix C.5), but the step length used is also decreased to $1.25 \cdot 10^{-6}$ seconds. It is also seen that the power distribution is very slanting - by more than a factor 100 between regions 1 and 3.

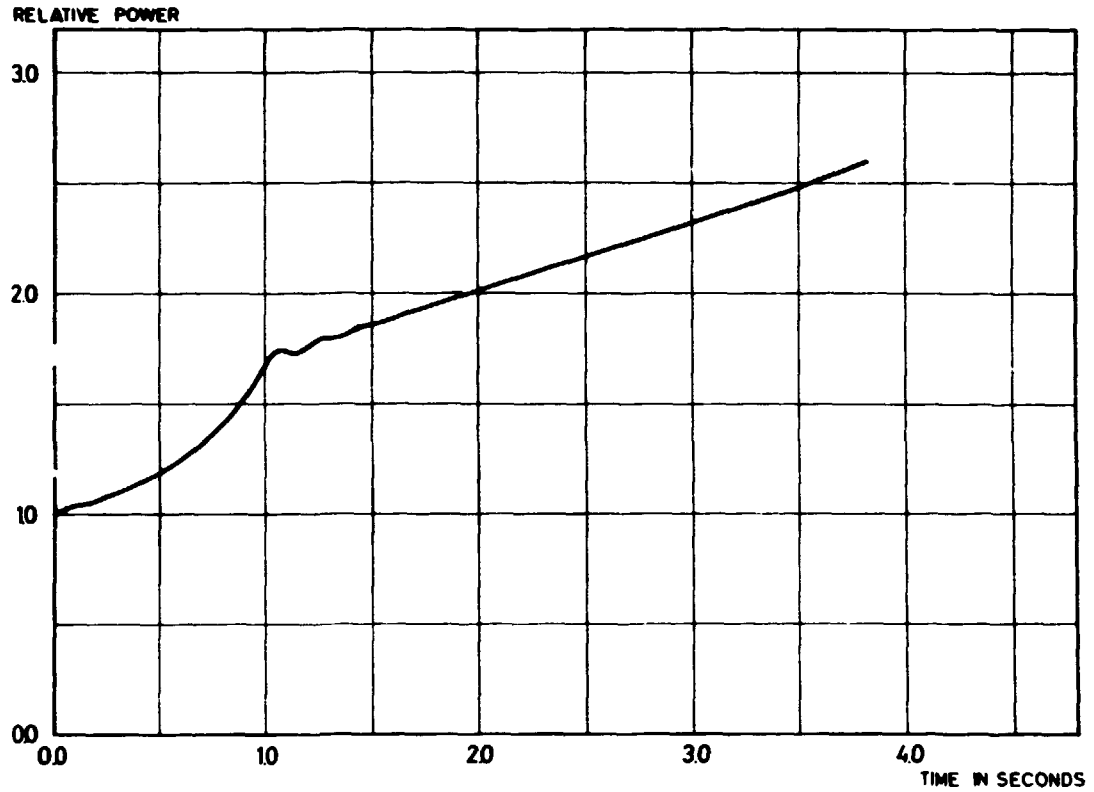


Fig. 5.10. The average power versus time for the slab reactor model, when the thermal absorption cross section in region 1 is linearly decreased by 1% in 1.0 second.

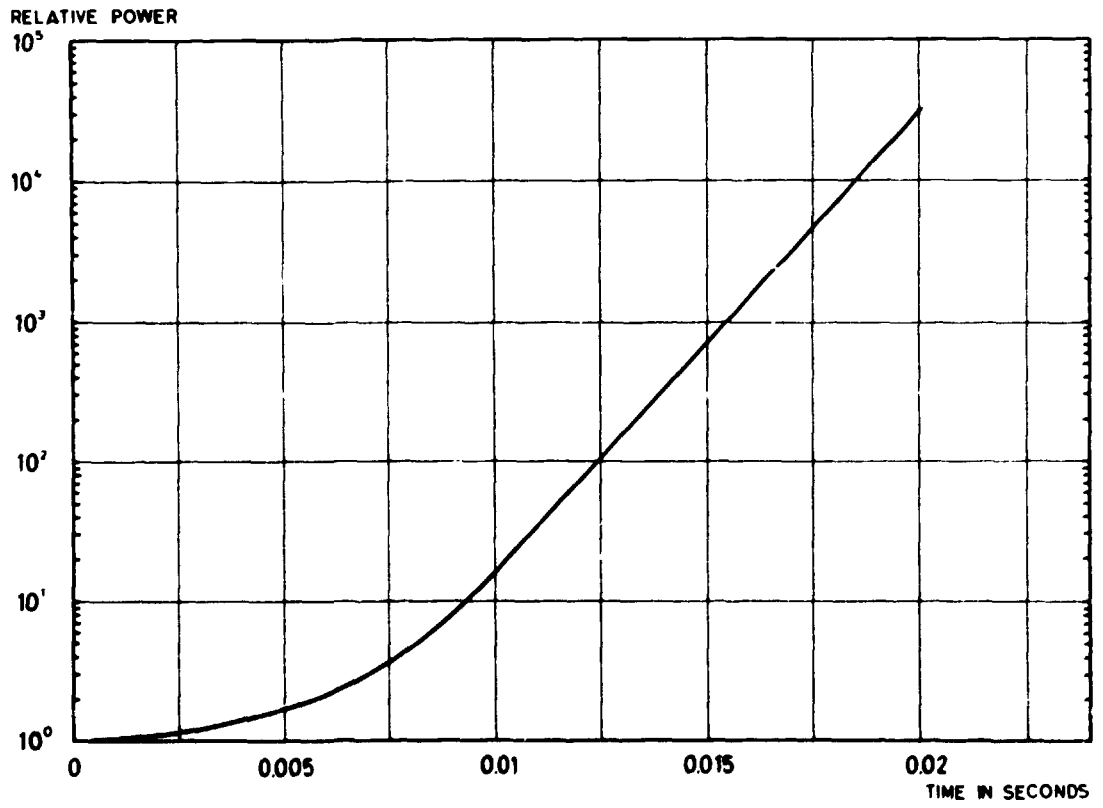


Fig. 5.11. The average power versus time for the slab reactor model, when the thermal absorption cross section in region 1 is linearly decreased by 5% in 0.01 second.

The last transient is the same as the preceding one, but now the average neutron velocities are 100 times faster. It results in a very fast transient and therefore it is followed only to 0.005 seconds. The step length quickly decreases to the smallest allowed value and the program takes a lot of time steps without fulfilling the convergence criterion; however, it does not seem to affect the accuracy of the solution. The fast increase in the power is shown in Fig. 5.12.

Table 5.13 shows the parameters which were used for the calculations. The computer times used are quite large, but it may not be necessary to take so many time steps as were used here.

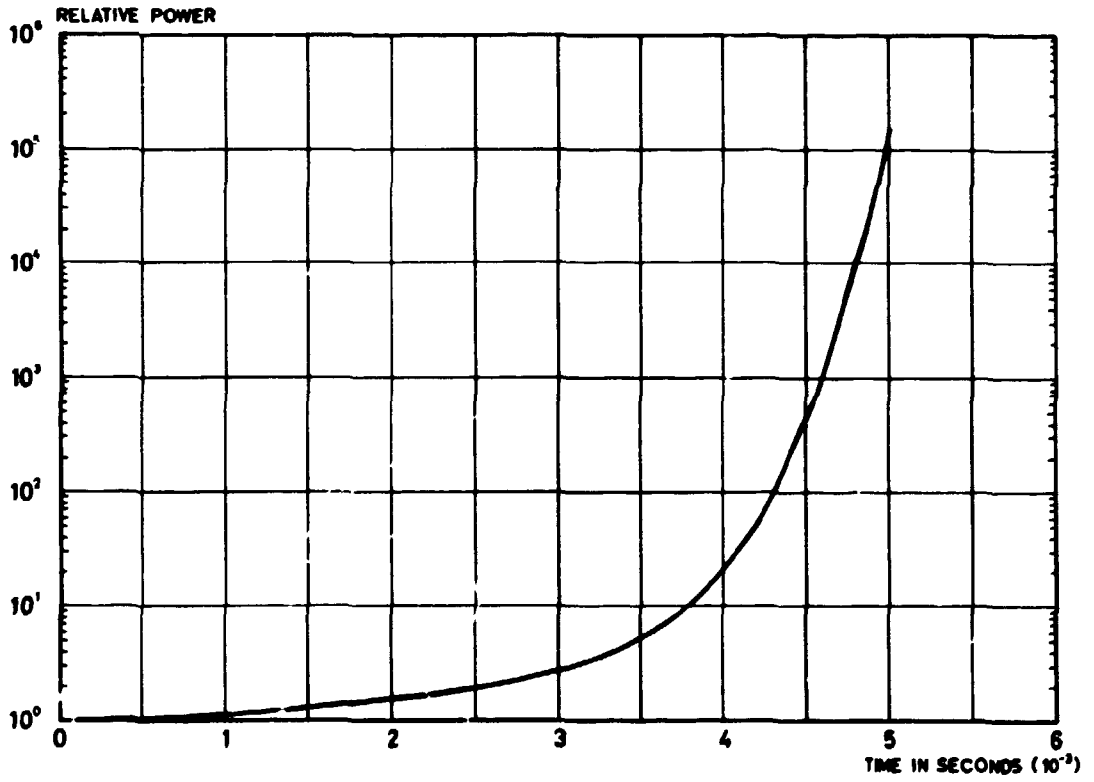


Fig. 5.12. The average power versus time for the slab reactor model, when the thermal absorption cross section in region 1 is linearly decreased by 5% in 0.01 second. The neutron velocities are 100 times larger than in Fig. 5.11.

Table 5.13. Main parameters for the four transients in the infinite slab reactor model.

	Transient			
	1	2	3	4
Mode	2	2	2	2
Maximum number of inner iterations	20	20	20	20
Minimum number of time steps before step length doubling	5	5	5	5
Tolerance on flux DF	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Initial time step	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Minimum allowed step	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Minimum step used	10^{-5}	10^{-5}	$1.25 \cdot 10^{-6}$	$6.25 \cdot 10^{-7}$
Maximum step used	0.0819	0.164	$4 \cdot 10^{-5}$	10^{-5}
CPU-time in seconds	134	265	2620	~2900

5.6. The TWIGL two-dimensional seed-blanket reactor kinetics model

A description of the benchmark, which is found in Smith (1979), is given in Appendix B.6. Two transients are initiated either by decreasing the absorption cross section in region 1 by 0.0035 cm^{-1} either as a step perturbation or as a ramp perturbation in 0.2 seconds.

Steady-state solutions for this problem are obtained by fourth-order local flux expansion order and various transverse leakage orders. The results are shown in Table 5.14 and they are very much alike. The steady-state power distribution is shown in Appendix C.6 and a diagram of it is shown in Fig. 5.13. The nodal size used is 8.0 cm.

The transients are calculated with the parameters shown in Table 5.15. The orders used are 4,1, but for this example there are no significant differences in the results between those obtained by orders 4,1 and orders 4,2. The convergence criterion in the inner iterations is the loose one with the sum of fluxes, but on the contrary the demand on the flux can be made stricter. For both transients a flux tolerance of 10^{-6} has been used, but for the example with the ramp perturbation a tolerance of 10^{-4} has been attempted.

Table 5.14. Steady-state results for the TWIGL example.

Orders	Mode	Inner iterations per outer	Outer iterations	Eigenvalue	CPU sec
2,0	3	5	9	0.90608	-
4,2	2	5	+24	0.91312	24.7
4,1	0	10	29	0.91318	37.0
4,1	2	10	25	0.91315	34.1
4,2	0	10	29	0.91318	37.6
4,2	2	10	29	0.91318	37.2

2 inner iterations per A-coefficient calculation
Tolerance on eigenvalue: DL = 10^{-6}
Tolerance on flux: DF = 10^{-4}

The initial step length could seem very small, but it is necessary to obtain an accurate solution. With a larger initial step length a number of rather large steps would be taken before the code "discovers" the perturbation, which would result in an incorrect delay of the transient.

The calculated average powers are tabulated in Table 5.16 and the two fine solutions are drawn in Fig. 5.14. The two fine

Table 5.15. Main parameters for the two transients with the two-dimensional TWIGL benchmark.

	Perturbation	
	Ramp	Step
Order	4,1	4,1	4,1
Mode	0	0	0
Maximum number of inner iterations	20	20	20
Minimum number of time steps before step length doubling	3	3	3
Tolerance on flux	10^{-6}	10^{-4}	10^{-6}
Initial time step	10^{-5}	10^{-5}	10^{-5}
Minimum allowed step	10^{-6}	10^{-6}	10^{-6}
Minimum step used	$5 \cdot 10^{-6}$	10^{-5}	$5 \cdot 10^{-6}$
Maximum step used	0.0410	0.0819	0.0819
CPU-time in seconds	2435	269	2513

Table 5.16. Average power for the TWIGL-benchmark as function of time.

	Ramp-perturbation			Step-perturbation	
Time sec	Fine NEM	Coarse NEM	Quandry	NEM	Quandry
0.00	1.000	1.000	1.000	1.000	1.000
0.05	1.131	1.070		2.065	
0.10	1.316	1.267	1.307	2.081	2.061
0.15	1.577	1.470		2.088	
0.20	1.972	1.835	1.957	2.096	2.078
0.25	2.082	2.200		2.104	
0.30	2.090	2.230	2.074	2.111	2.095
0.35	2.098	2.238		2.117	
0.40	2.105	2.249	2.092	2.124	2.113
0.45	2.113	2.257		2.132	
0.50	2.122	2.265	2.109	2.141	2.131

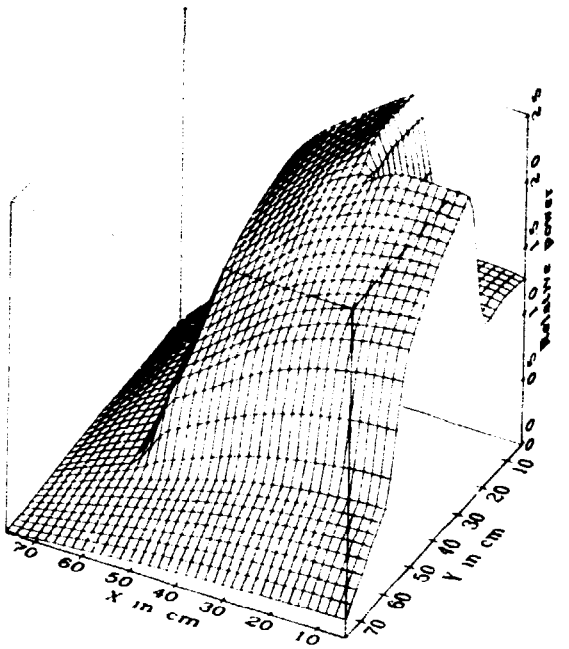


Fig. 5.13. The steady-state power distribution for one quarter of the core for the two-dimensional TWIGL benchmark.

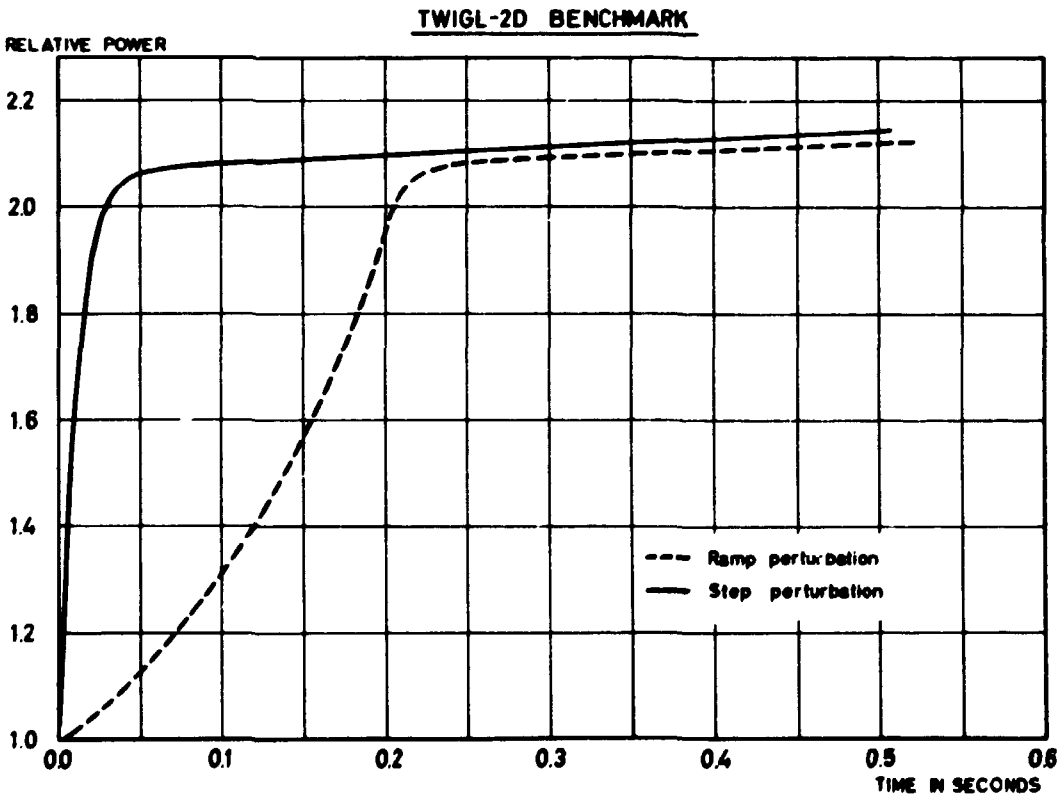


Fig. 5.14. The average power versus time for the two-dimensional TWIGL benchmark for respectively ramp- and step-perturbation.

solutions are in good agreement with the QUANDRY-solutions, but they also take long times to produce. The solution with the coarse flux tolerance is nearly 10 times faster, but it is not so accurate with about 8% deviation from the reference. The normalized power distributions to various times are printed in Appendix C.6 and though the average power increases with more than a factor of two the largest changes in the power distribution are less than 5%.

5.7. The LMWLWR transient problem

This benchmark, which is a very simple model of a LWR was first decribed by Langenbuch, Maurer and Werner (1977) and is therefore called the LMWLWR problem. The benchmark is described in Appendix B.7 and the static solution is shown in Appendix C.7. The static solution is obtained with orders 4,2, and a nodal size of 20 cm in all directions; it took 24 outer iterations and 161 seconds of CPU-time.

The transient caused by some movements of the control rods is calculated by two different methods. The method used in the preceding, which yields very accurate but also very slow results, is too slow for this three-dimensional calculation. The first method (A) simply uses constant time steps independent of the convergence of the fluxes in the inner iterations. In the problems described in Paragraphs 5.5 and 5.6 a very small initial step length was demanded, but this is circumvented by provoking a lot of inner iterations (200) in the first step.

The method is tested with various step lengths from 0.1 second to 2.0 seconds and the results are tabulated in Table 5.17. The reference solution is obtained with QUANDRY (Smith, 1979), and in Table 5.18 the deviations to the NEM-solutions are tabulated. With time steps = 0.1 and 0.25 sec the solutions are quite good within, respectively, 1% and 4%. A time step of 0.50 sec gives an acceptable solution, but a rather large overshoot has occurred. Larger time steps give unacceptable solutions.

The second method uses variable time steps, but a much less restrictive method for determining the step size is used. The convergence criterion used in the step size determination is moved from the inner to the outer iterations. The new convergence criterion for the outer iterations is based on the average power in two subsequent time steps

$$\frac{|P(t) - P(t-\Delta t)|}{P(t)} < DP$$

The criterion is not really a convergence one, but it prevents the solution from changing too much between each step. If the criterion is unfulfilled, the step length is halved unless the minimum step length is already used. The step length is doubled when the relative power difference is less than half the power tolerance and a number of time steps (specified in the input) have been taken since last time-step change. The tolerance in the power difference, DP must be set suitably and in this example DP = 0.01 and 0.03 have been used.

The results are also tabulated in Tables 5.17 and 5.18, and they are not quite as good as those obtained with method A. All the solutions are graphically depicted in Fig. 5.15 and the solutions obtained with method B have a rather peculiar behaviour. Both solutions have some sharp peaks, which cannot be attributed to any physical reason. This behaviour is caused by numerical overshooting that would result in wild oscillations if the step length were not drastically decreased. In the coarse case the step length is decreased to 0.0125 seconds and in the fine case it is decreased to $7.8 \cdot 10^{-4}$ (minimum allowed) sec which would bring the oscillations under control, but could not avoid the peaks. Method A may be preferred for this example.

Table 5.17. Parameters and results for the LMWLWR transient problem.

Program	A	A	A	A	A	B	B	QUANDRY
Orders	4,2	4,2	4,2	4,2	4,2	4,2	4,2	
Mode	2	2	2	2	2	2	2	
Maximum no of inner it.	20	20	20	20	20	20	20	
Initial step length	0.1	0.25	0.5	1.0	2.0	0.1	0.1	
Minimum step length						0.001	0.001	
Tolerance	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	$3 \cdot 10^{-2}$	
P(0.0 s)	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0
P(5.0 s)	167.8	163.7	155.9	154.4		167.8	165.4	167.1
P(10.0 s)	197.4	197.7	181.2	161.7	158.7	198.1	192.8	196.9
P(15.0 s)	234.9	232.2	221.8	179.6		234.4	222.5	
P(20.0 s)	251.1	253.8	257.6	205.9	169.4	255.3	254.5	250.3
P(25.0 s)	236.2	244.2	266.3	230.9		262.0	280.0	
P(30.0 s)	199.2	203.8	230.2	236.5	183.0	299.0	293.0	201.0
P(35.0 s)	156.2	156.0	164.2	216.9		166.7	292.0	
P(40.0 s)	117.9	117.4	110.0	178.0	181.3	126.1	263.0	119.0
P(45.0 s)	90.5	89.7	84.5	114.0		95.8	156.8	
P(50.0 s)	73.7	71.5	72.1	53.1	138.2	77.0	101.1	74.1
P(55.0 s)	63.7	64.0	61.8	15.0		66.7	96.7	
P(60.0 s)	56.4	56.1	55.3	12.2	71.6	59.0	80.1	56.9
P_{max}	253.0	255.9	268.0	237.9	186.3	262.0	293.9	
$T(P_{max})$	21.2	21.75	235	29.0	34.0	25.0	32.6	
CPU	8062	3248	1705	889	460	7212	2583	

Table 5.18. Deviations from the QUANDRY reference solution in % for the LMWLWR transient problem.

Method	A	A	A	A	A	B	B
Step length	0.1	0.25	0.50	1.0	2.0		
Tolerance						10^{-2}	$3 \cdot 10^{-2}$
T=5.0 s	0.4	-2.0	-6.7	-7.6		0.4	-1.0
T=10.0 s	0.3	0.4	-8.0	-17.9	-19.4	0.6	-2.1
T=20.0 s	0.3	1.4	2.9	-17.7	-32.3	2.0	1.7
T=30.0 s	-0.9	1.4	14.5	17.7	-9.0	13.9	45.8
T=40.0 s	-0.9	-1.3	-7.6	49.6	52.4	6.0	121.0
T=50.0 s	-0.5	-3.5	-2.7	-28.3	86.5	3.9	36.4
T=60.0 s	-0.9	-1.4	-2.8	-78.3	25.8	3.7	40.8

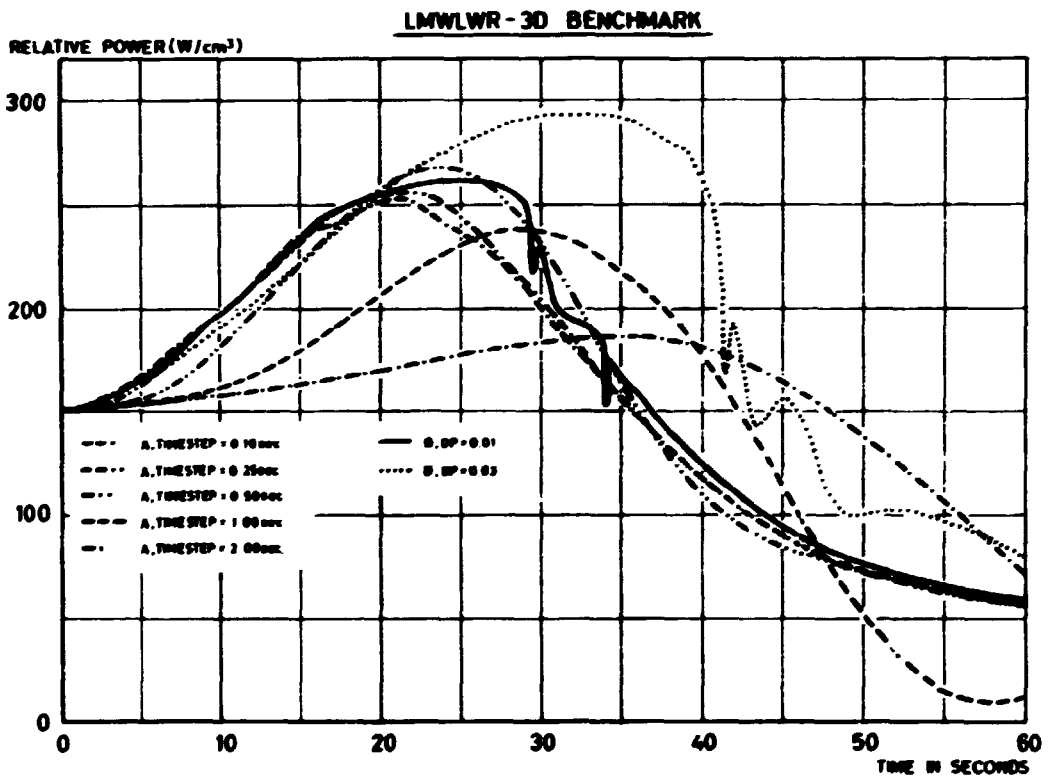


Fig. 5.15. The average power density as function of time for the three-dimensional LMWLWR benchmark. Various steplengths and steplength selection strategies are used.

5.8 The LRABWR two-dimensional benchmark

The LRABWR benchmark is a simplified model of a BWR in two energy groups. The reactor has a two-zone core containing 312 fuel elements, each having a width of 15 cm. The core is reflected by 30 cm of pure water and the control rods are represented as smeared absorbers in four adjacent fuel assemblies. The existence of the control rods causes severe local flux perturbations and the problem is rather difficult to solve. The benchmark is completely described in Appendix B.8.

The steady-state solutions are shown in Table 5.19., where the reference solutions are obtained by QUANDRY (Smith, 1979). The results under 1 and 2 are for different orders and modes, while result 3 is with orders 2,0 at the start and orders 4,2 at the end. Result 4 is with finer nodes and the only difference between 2 and 4a is the core geometry. It is seen that the com-

Table 5.19. Steady-state results for the LRABWR two-dimensional benchmark.

	1	2	3a	3b	4a	4b	4c	5
Rod position	in	in	in		in			out
Geometry	1/4	1/4	1/8		1/8			1/4
Node size in cm^2	15^2	15^2	15^2		15^2	7.5^2	3.75^2	15^2
Orders	4,1	4,2	2,0	4,2	4,2			4,1
Inner it/outer	17	10	17		10			17
Mode	0	2	3	2	2			2
Outer iterations	37	40	7	+26	44	+14	+9	40
Eigenvalue, λ	.99626	.99619	.99598	.99619	.99619	.99630	.99630	1.01508
$\Delta\lambda \cdot 10^5$	-10	-17	-38	-17	-17	-6	-6	-41
ΔP_{max} %	2.96	3.89	59.9	4.13	3.89	3.02	2.55	6.41
ΔP_{av} %	0.86	1.13	11.0	1.26	1.17	0.93	0.78	1.62
CPU seconds	163	154	8	+54	83	+76	+211	148

puter time is nearly proportional to the number of nodes. The final results are for the situation with the control rod 12 withdrawn in the cold state. The normalized power distributions are tabulated in Appendix C.8 and shown in Figs. 5.16 and 5.17.

The transient of this benchmark is induced by the rapid withdrawal of a control rod. A very simple feedback mechanism is built-in in this problem. The feedback model is specified by two relations:

adiabatic heatup:

$$\frac{\partial}{\partial T} T(\bar{r},t) = \alpha \cdot [\Sigma_{f1}(\bar{r},t) \cdot \phi_1(\bar{r},t) + \Sigma_{f2}(\bar{r},t) \cdot \phi(\bar{r},t)]$$

and Doppler feedback:

$$\Sigma_{a1}(\bar{r},t) = \Sigma_{a1}(\bar{r},0) \cdot [1+\gamma(\sqrt{T(\bar{r},t)} - \sqrt{T_o})]$$

where α, γ and T_o are known constants (see Appendix B.8 for a complete description).

The problem has been solved by the three methods described above and they are compared with a QUANDRY-solution. The average power as function of time is tabulated in Table 5.20 and graphically shown in Fig. 5.18. The average power increases by about 10 orders of magnitude, which is a very hard test for a computer program for solving a numerical differential equation. The results from the three different methods differ very much from each other, but the shape of the curves in Fig. 5.18 are similar, but time-displaced. The temperature is integrated by the backward Euler method, which also introduces some errors.

The normal NEM method is stopped after about 1.7 second due to exceeded process time. The method with constant time steps is quite close to the normal method while method B has a time delay of some tenths of a second which in its turn causes a higher power peak. In Appendix C.8 the normalized power peak distribution from method A is given at various times, and it is also shown in Fig. 5.19.

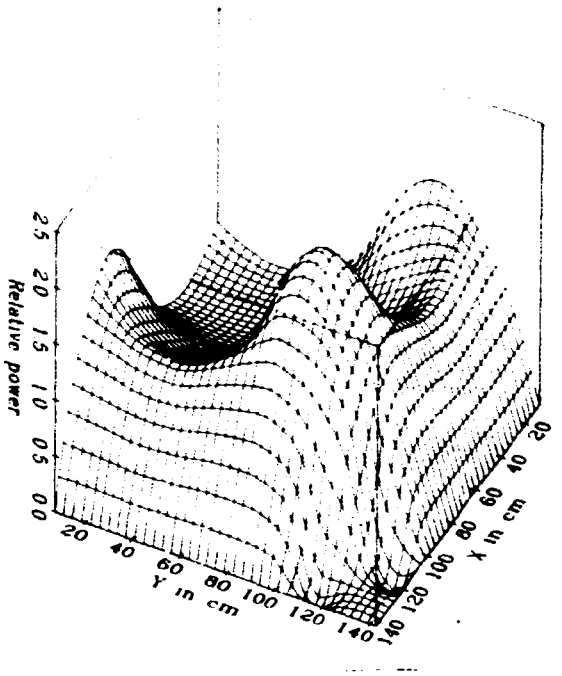


Fig. 5.16. The power distribution for one quarter of the core for the two-dimensional LRABWR benchmark with the control rods in.

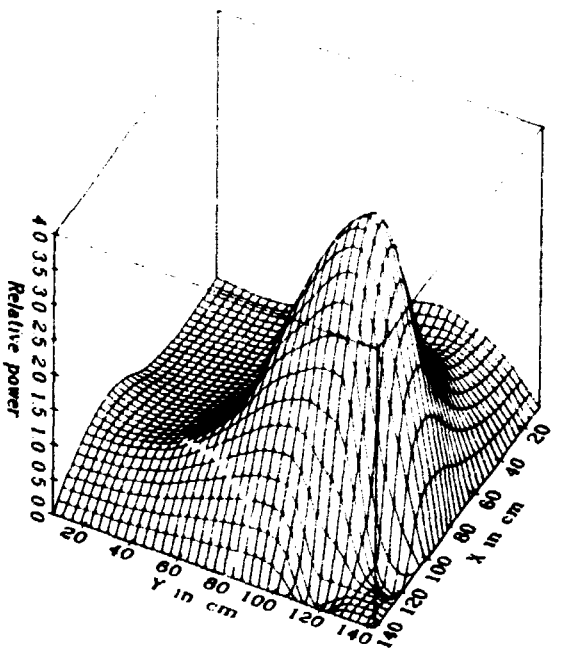


Fig. 5.17. The power distribution for one quarter of the core for the two-dimensional LRABWR benchmark with the control rods out in cold condition. (There is no Doppler feddback).

Table 5.20. Kinetic results for the LRABWR two-dimensional benchmark.

Program	NEM	NEM, A	NEM, B	QUANDRY
Orders	4,1	4,1	4,1	
Mode	2	2	2	
Maximum no. of inner it.	20	20	20	
Initial step length	0.002	0.001	0.002	
Minimum step length	10^{-4}	-	10^{-5}	
Tolerance	0.01	0.1	0.1	
P(0.0 s)	10^{-6}	10^{-6}	10^{-6}	10^{-6}
P(0.4 s)	$1.36 \cdot 10^{-6}$	$1.38 \cdot 10^{-6}$	$1.30 \cdot 10^{-6}$	$1.38 \cdot 10^{-6}$
P(0.8 s)	$2.53 \cdot 10^{-6}$	$2.94 \cdot 10^{-6}$	$1.90 \cdot 10^{-6}$	$3.05 \cdot 10^{-6}$
P(1.2 s)	$2.89 \cdot 10^{-5}$	$2.34 \cdot 10^{-4}$	$3.05 \cdot 10^{-6}$	$7.35 \cdot 10^{-4}$
P(1.4 s)	12.2	9.11	$4.18 \cdot 10^{-6}$	720
P(1.6 s)	165	154	$7.10 \cdot 10^{-6}$	
P(1.8 s)	-	341	$3.45 \cdot 10^{4}$	
P(2.0 s)	-	1093	163	79.8
P(2.5 s)	-	139	90.4	
P(3.0 s)	-	74.1	44.0	97.4
$P_{max,1}$	6797	7447	$3.45 \cdot 10^{4}$	5749
$T(P_{max,1})$	1.477	1.500	1.801	1.436
$P_{max,2}$	-	1098	177.4	861
$T(P_{max,2})$	-	2.006	1.903	
used step:				
Minimum	$6.25 \cdot 10^{-5}$	-	$2.5 \cdot 10^{-4}$	
Maximum	0.032	-	0.064	
CPU seconds	80000	26473	5721	

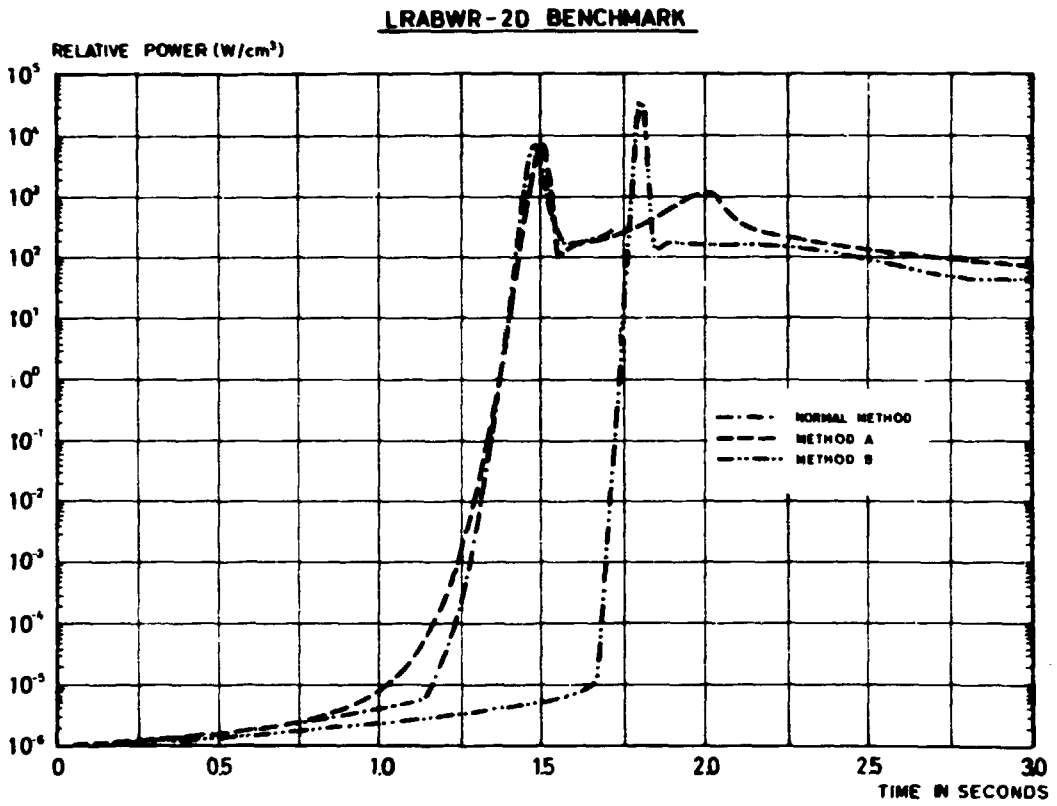
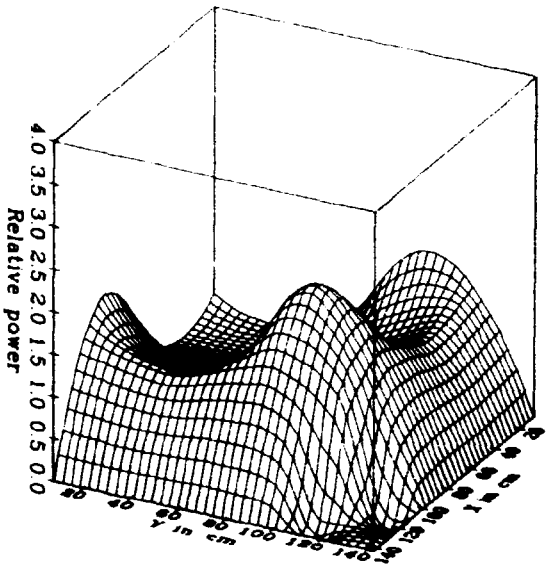
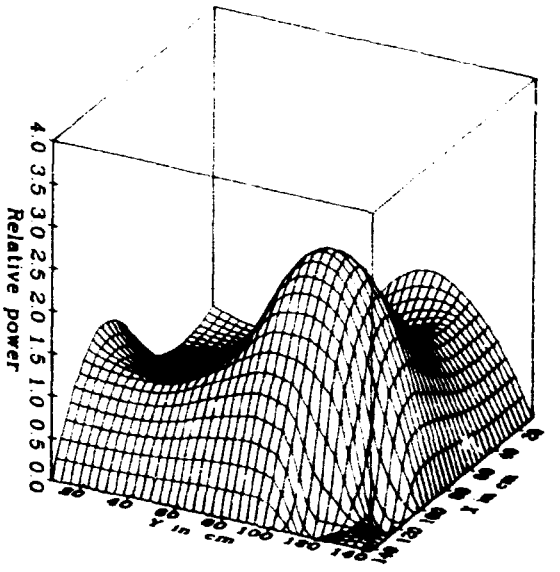


Fig. 5.18. The average power versus time for the two-dimensional LRABWR benchmark with a simple Doppler feedback model. Various steplength strategies are used.

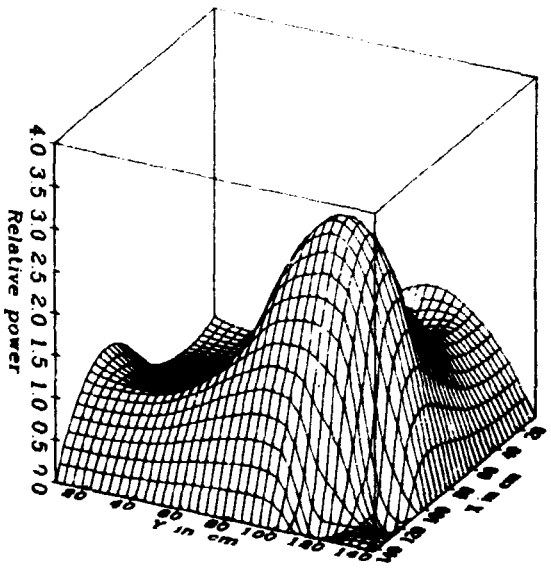
Time = 0.0 sec



Time = 0.8 sec



Time = 1.2 sec



Time = 1.4 sec

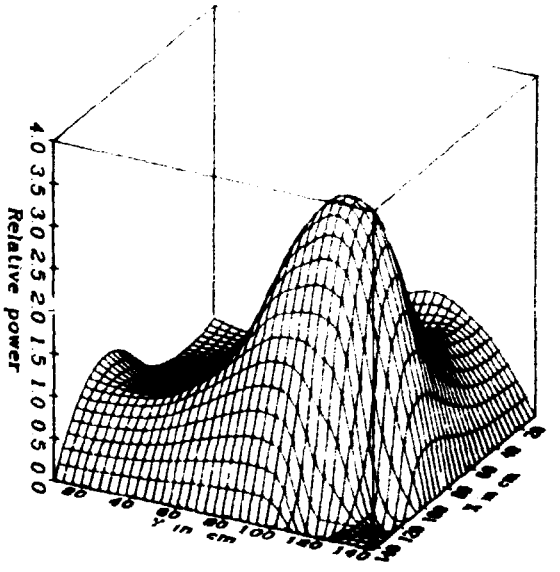
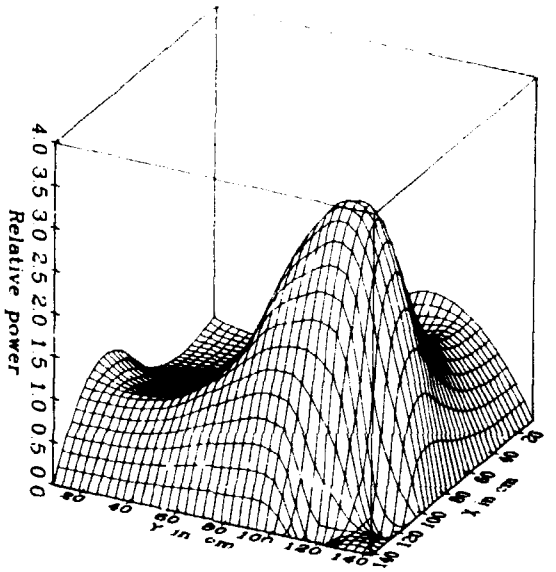
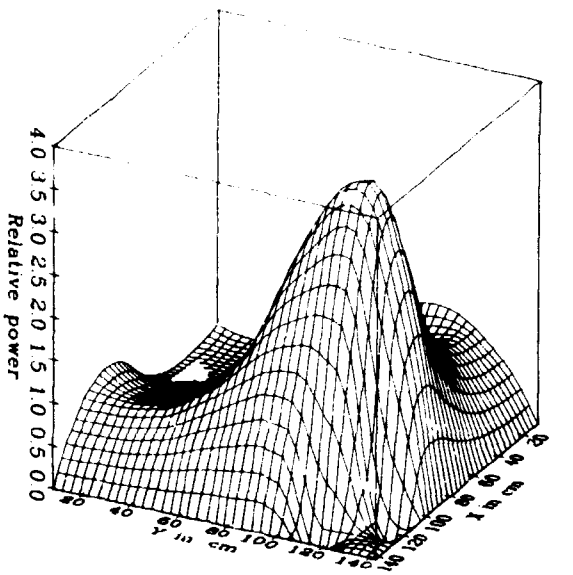


Fig. 5.19a. The power distribution for one quarter of the core for the two-dimensional LRABWR benchmark to various times during the transient.

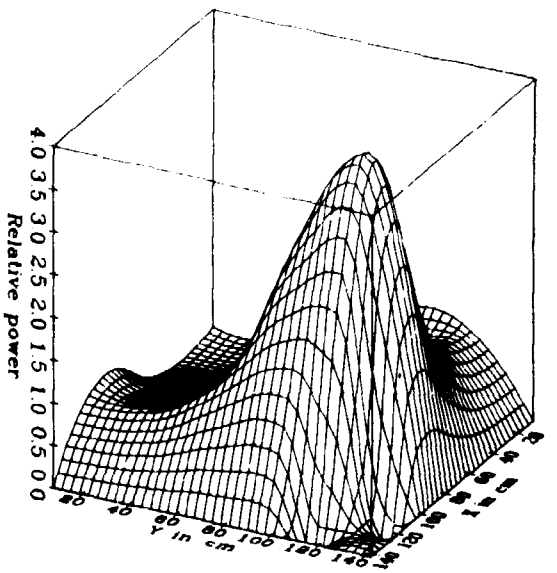
Time = 1.6 sec



Time = 1.8 sec



Time = 2.0 sec



Time = 3.0 sec

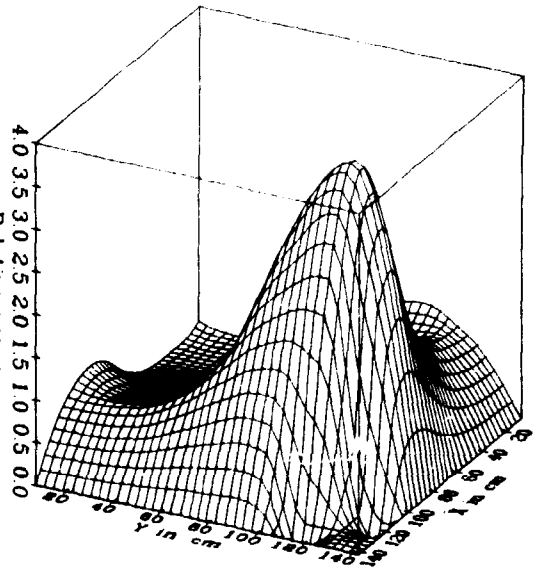


Fig. 5.19b. The power distribution for one quarter of the core for the two-dimensional LRABWR benchmark to various times during the transient.

5.9 The LRABWR three-dimensional benchmark

The preceding two-dimensional benchmark is extended to three dimensions with a core height of 300 cm and an axial reflector of 30 cm water in each end. Only static results are given here and they are summarized in Table 5.21, where the results under 2 and 5 are obtained with the method described in Section 5.1.3. It is seen that there are great savings to be had by using low order in the beginning without losing accuracy. The errors are calculated by comparison with a QUANDRY-solution.

Table 5.21. Results for the LRABWR three-dimensional benchmark.

	1	2a	2b	3	4	5a	5b
Rod position	in	in	-	in	out	out	-
Geometry	1/8	1/8		1/4	1/4	1/4	-
Node size in cm^3	$15^2 \times 30$	$15^2 \times 30$	-	$15^2 \times 30$	$15^2 \times 30$	$15^2 \times 30$	-
Orders	4,1	2,0	4,2	4,1	4,1	2,0	4,2
Inner it/ outer	10	17	-	17	17	17	-
Mode	2	3	2	0	0	3	2
Outer it.	77	11	+20	42	43	12	+34
Eigenvalue λ	.99617	.99600	.99617	.99628	1.01512	1.01303	1.01500
$\Delta\lambda \cdot 10^5$	-22	-39	-22	-8	-37	-246	-49
ΔP_{max} %	1.95	24.43	3.42	2.11	2.45	75.91	3.63
ΔP_{av} %	0.61	-	-	0.64	0.67	16.43	0.99
CPU sec	1690	128	+454	2123	2155	253	+1033

6. CONCLUSION

In the preceding chapters I have described the development of a program based on the Nodal Expansion Method. The program is divided into two main parts: one for the steady-state calculations and another for the dynamic calculations. The subroutines used in the two parts are very much alike, so the steady-state routine is described most intensively.

The program has been tested on a number of examples with good results. The calculations can be made with various approximations and acceleration methods. The results for calculations with high orders (fourth-order local flux expansion order and first- or second-order transverse leakage expansion) are very accurate when compared with other calculations (finite difference, etc.). The calculations are also rather fast compared with other programs for the same accuracy. The nodal code ANTI is faster, but it also uses only a 1 1/2-group method, while NEM uses 2 groups or more.

It has also been tried to make calculations on a real reactor and then compare the results with measurements. This attempt was more unsuccessful - the calculated power distribution disagrees significantly from the measurements. The cause of this discrepancy was investigated, and the main indefiniteness lies around the boundary conditions at the reflector.

The reflector can be represented in two ways, either by some albedo-matrices or by cross sections and diffusion coefficients with an outer boundary condition of zero flux or zero incoming currents. The albedo-matrix representation is the most favourable with respect to the number of variables; however, the solution is changed for different node sizes with the same albedos. With the reflector represented as a reflector zone of additional nodes the solutions are nearly independent of node size, but the number of nodes increases a lot; besides it is a problem to calculate homogenized and condensed material constants for the reflector. Sev-

eral attempts have been made to make a good reflector representation, but with no great success. The reflector representation or perhaps the whole homogenization procedure has to be researched further.

The program has also been tested on dynamic problems, but with not such great success relative to the steady-state calculations. The time steps that are to be used for accurate calculations are very small slowing down the code. Several time step selection strategies have been tried, and some of them work rather well, but depend on the problem to be solved.

The program uses a direct method to solve the time-dependent problems, which means that the time-integration and the space-solutions are fully integrated, but perhaps it is necessary and also satisfatory to use a quasi-static method. Further investigations must show this, and besides a hydraulic model has to be built in, which may move the problems to other places in the assembled program. The hydraulic part can be taken from the ANTI-program, and in ANTI the hard points are in this part.

The examples I have used are mostly commonly known benchmarks and the reference solutions are calculated with other codes. Someone could claim that it is very well to compare the results with the results obtained with other codes, but if they all are wrong, they may have little to do with the real world. The statement is partly true. If the program describes the whole reactor with hydraulic feedback, pumps, turbines, etc. it is true, but if a single part of this program complex is taken out, it is nearly impossible to compare the results from this part with measurements. The program part has to be set in the right surroundings and often these surroundings disturb the results completely, which also are seen by the example with the typical PWR. In that example, the hydraulic part was coupled off, the homogenization was done, etc., but nevertheless a problem with the neutronic boundary conditions disturbs the solution. Therefore the only way to verify a program often is to compare with the results from other codes.

As a final remark, some remaining problems, improvements and possibilities for the Nodal Expansion Method and the NEM-program at Risø National Laboratory will be outlined.

1. Introduction of hydraulic, either NEM in ANTI or reversed.
2. Improvement of the time step selection algorithm.
3. Investigations for a better reflector representation.
4. Introduction of Koebke's discontinuity factors.
5. Introduction of a possiblity for node interior cross section variation, especially around control rods.

REFERENCES

Argonne Code Center (1977). Benchmark Problem Book. Numerical Determination of the Space, Time, Angle, or Energy Distribution of Particles in an Assembly. Argonne National Laboratory - 7416.

BABALA, D., BECH, N., HAUGSET, H. (1971). ANDYCAP - Vol. 1 - Theory SRD440.

BØRRESEN, S. (1981). Experience, Status and Advanced Applications of PRESTO. Proceedings of the International Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems. München, 27-29. April 1981, p. 283-297.

CADWELL, W.R. HENRY, A.F., and VIGILOTTI, A.J. (1964). WIGLE - A Program for the Solution of the Two-group Space-time Diffusion Equation in Slab Geometry. Westinghouse Electric Corporation, Bettis, Atomic Power Laboratory Report WAPD-TM-416.

CADWELL, W.R. (1967). PDQ-7. Reference Manual. WAPD-TM-678, Bettis Atomic Power Laboratory.

DORNING, J. (1979). Modern Coarse-mesh Methods - A Development of the '70's. Proceedings of an ANS Meeting: Computational Methods in Nuclear Engineering. Williamsburg, Virginia, April 23-25, 1979, p. 3.1-3.31.

EL-WAKIL, M.M. (1962). Nuclear Power Engineering. (McGraw-Hill Book Company, New York). 556 pages.

FINNEMANN, H. (1975). A consistent Nodal Method for the Analysis of Space-Time Effects in large LWR's. In: Proceedings of the Joint NEACRP/CSNI Specialists' Meeting on New Developments in Three-Dimensional Neutron Kinetics and Review of Kinetics Benchmark Calculations, Garching (Munich) January 22-24, 1975, 133-170.

FINNEMANN, H., BENNEWITZ, F. and WAGNER, M.R. (1977). Interface current techniques for multidimensional reactor calculations. Atomkernenergie 30, 123-128.

FRANKE, H.P. (1977). Anwendung der Methode der finiten Elemente bei der Berechnung drei-dimensionaler reaktor-physikalischer Probleme. Atomkernenergie 30. p. 88-94.

GLASSTONE, S. und EDLUND, M.C. (1961). Kernreaktortheorie. Springer-Verlag, Wien, Deutschen Ausgabe. 341 pages.

JAGANNATHAN, V. (1983). Evaluation of the finite-element-synthesis model using the 3-D finite-element technique. Annals of nuclear energy 10 pp. 569-578.

KOEBKE, K. (1978). A New Approach to Homogenization and Group Condensation. In: Homogenization Methods in Reactor Physics, held in Lugano, 13-15 November 1978 (IAEA-TECDOC-231) p. 303-323.

KRISTIANSEN, G.K. (1976). The finite-difference neutron diffusion programme TWODIM. Risø-M-1891.

KRISTIANSEN, G.K. (1977). The two-dimensional finite-difference programme TVEDIM. Risø-M-1919.

LANGENBUCH, S., MAURER, W. and WERNER, W. (1977a). Coarse-mesh Flux-expansion Method for the Analysis of Space-Time Effects in Large Light Water Reactor Cores. Nuclear Science and Engineering 63 p. 437-456.

LANGENBUCH, S., MAURER, W. and WERNER, W. (1977b). High-order Schemes for Neutron Kinetics Calculations, Based on a Local Polynomical Approximation. Nuclear Science and Engineering 64, p. 508-516.

LARSEN, A.M.H. (1973). CRS: A Code to produce Multigroup Neutron Cross Sections for Reactor Physics Calculations. Risø-M-1568.

LARSEN, A.M.H. (1980). The Three-Dimensional PWR Transient Code ANTI; Rod Ejection Test Calculation, Risø-M-2209.

LARSEN, A.M.H. (1983). Personal communication.

LARSEN, H. (1971). SYNTRON, A Three-dimensional Flux Synthesis Programme. Risø-M-1346.

LAURIDSEN, B. (1977). CRSIQ: A Cluster Programme to produce Multigroup Cross Sections. RP-12-77. 17 pp. Risø National Laboratory.

LAWRENCE, R.D. (1979). A Nodal Green's Function Method for Multidimensional Neutron Diffusion Calculations. Dissertation (Ph.D.), University of Illinois, Urbana Il.

LIEBEROTH, J. (1977). An approximation of finite-difference methods for diffusion problems by a coarse-mesh method. Atomkernenergie 30, p. 103-106.

LINDSTRØM JENSEN, K.E. (1971). Development and Verification of Nuclear Calculation Methods for Light Water Reactors. Risø Report No. 235. 161 p.

MICHEELSEN, B. and NELTRUP, H. (1973). The 3D-IAEA Benchmark Problem. Risø-M-1572.

MISFELDT, I. (1975). Solution of the Multigroup Neutron Diffusion Equation by the Finite Element Method Risø-M-1809.

NAKATA, H. and MARTIN, W.R. (1983). The finite element response matrix method. Nuclear Science and Engineering 85, 289-305.

NIELSEN, E.F. and LARSEN, A.M.H. (1980). Input Description for the Three-dimensional PWR Transient Code ANTI. Risø-M-2256.

PEDERSEN, J. (1969). Calculation of Heterogeneous Constants for Cylinders and Slabs. Risø-M-850.

RYDIN, R.A. and SULLIVAN T.M. (1978). A new approach to the QUABOX-CUBBOX coarse-mesh methods. Advances in Reactor Physics.

SCHOUGAARD, B.F. (1979). A short Description of the Programme NOTAM. Risø RP-19-79.

SIEWERS, H. and JAGER, W. (1977). Comparison of coarse-mesh methods for benchmark and realistic reactor problems. Atomkernenergie 30 p. 107-112.

SMITH, K.S. (1979). An analytical nodal method for solving the two-group, multidimensional, static and transient neutron diffusion equation. Masters thesis, Massachusetts Institute of Technology.

THORLAKSEN, B. (1981). Construction of PWR Nuclear Cross Sections for Transient Calculations. Test of ANTI Program Against TWODIM. Risø-M-2264.

VONDY. D.R., FOWLER, T.B. and CUNNINGHAM, G.W. (1977). VENTURE: A Code Block for Solving Multigroup Neutronic Problems Applying the Finite-Difference Diffusion-Theory Approximation to Neutron Transport, Version 2. ORNL-5062/R1.

WACHSPRESS, E.L., BURGESS, R.D. and BARON, S. (1962). Multichannel Flux Synthesis. Nuclear Science and Engineering 12, p. 381-389.

WAGNER, M.R., FINNEMANN, H., KOEBKE, K. and WINTHER, H.-J. (1977). Validation of the Nodal Expansion Method and the Depletion Program MEDIUM-2 by benchmark calculations and direct comparison with experiment. Atomkernenergie 30, 129-135.

WAGNER, M.R., KOEBKE, K. and WINTHER, H.-J. (1981). A Nonlinear Extension of the Nodal Expansion Method. In: Proceedings of the International Topical Meeting on Advances in Mathematical Methods of the Solution of Nuclear Engineering Problems. Münich, April 27-29, 1981. Vol. 2, 43-58.

WEISS, Z. (1977). Some Basic Properties of the Response Matrix Equations. Nuclear Science and Engineering 63, p. 457-492.

WERNER, W. (1975). Mathematical problems in three-dimensional reactor calculations. In: Proceedings of the joint NEACRP/ CSNI specialists meeting on new developments in three-dimensional neutron kinetics and review of kinetics benchmark calculations. Garching, 22-24 January, p. 91-105.

APPENDIX A

Basisfunctions

The basisfunctions which are used for the expansion of the one-dimensional flux are not some of the well-known functions, but they are nevertheless well defined.

The N'th basisfunction is a polynomial of N'th order

$$P_N(u) = \sum_{n=o}^{N} c_n \cdot u^n$$

with the following properties

1. Definition interval $I = [-1/2, +1/2]$

2. The coefficient to the highest degree term is equal to 1
   $c_N \equiv 1$

3. The integral over the definition interval is 0 for $N > 0$.

$$\int_{-1/2}^{+1/2} P_N(u)du = 0 \qquad N > 0$$

4. They are symmetrical around the interval centre

$$P_N(u) = P_N(-u) \qquad \text{for N even}$$

$$P_N(u) = -P_N(-u) \qquad \text{for N odd}$$

5. The function values in the extreme points are equal to 0

$$P_N(-1/2) = P_N(+1/2) = 0 \qquad \text{for } N > 2$$

6. The functions are orthogonal for N > 2

$$\int_{-1/2}^{+1/2} P_M(u)\cdot P_N(u)du = 0 \qquad M > 2,\ N > 2,\ M \neq N$$

The above properties unambiguously define the polynomials

$$P_0(u) = 1$$
$$P_1(u) = u$$
$$P_2(u) = u^2 - 1/12$$
$$P_3(u) = u^3 - 1/4\cdot u$$
$$P_4(u) = u^4 - 3/10\cdot u^2 + 1/80$$
$$P_5(u) = u^5 - 1/3\cdot u^3 + 1/48\cdot u$$

In Fig. A.1 are the basisfunctions of orders 0 - 5 shown, but with a different normalization to clarify the drawing

$$Q_0 = P_0$$
$$Q_1 = 2\cdot P_1$$
$$Q_2 = 6\cdot P_2$$
$$Q_3 = 12\cdot P_3$$
$$Q_4 = 30\ P_4$$
$$Q_5 = 72\cdot P_5$$

The alternative normalization is chosen

$$Q_N(1/2) = 1 \qquad N = 0,1,2$$
$$Q_N'(1/2) = 6 \qquad N > 2$$

The first derivative

$$P_0'(u) = 0$$
$$P_1'(u) = 1$$
$$P_2'(u) = 2u = 2\cdot P_1(u)$$
$$P_3'(u) = 3u^2 - \tfrac{1}{4} = 3\cdot P_2(u)$$
$$P_4'(u) = 4u^3 - \tfrac{3}{5}u = 4\cdot P_3(u) + \tfrac{2}{5}\cdot P_1(u)$$
$$P_5'(u) = 5u^4 - u^2 + \tfrac{1}{48} = 5\cdot P_4(u) + \tfrac{1}{2}\cdot P_2(u)$$

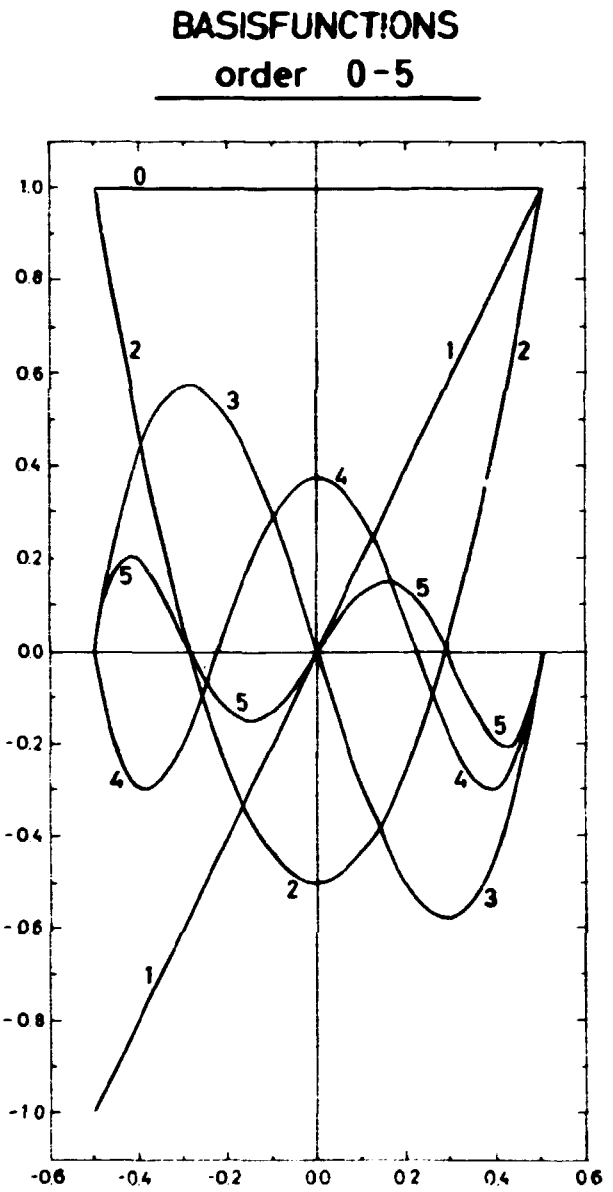


Fig. A.1. The basis-functions of orders 0-5.

The second derivative

$$P_0''(u) = 0$$

$$P_1''(u) = 0$$

$$P_2''(u) = 2 \cdot P_0(u)$$

$$P_3''(u) = 6 \cdot P_1(u)$$

$$P_4''(u) = 12 \cdot P_2(u) + \frac{2}{5} \cdot P_0(u)$$

$$P_5''(u) = 20 \cdot P_3(u) + 3 \cdot P_1(u)$$

The relevant integrals used in the weighted residual formula

$$\int_{-1/2}^{+1/2} P_M(u) \cdot P_N(u) du$$

M \ N	0	1	2	3	4	5
0	1	0	0	0	0	0
1	0	1/12	0	-1/120	0	-1/5040
2	0	0	1/180	0	- 1/2100	0
3	0	-1/120	0	1/840	0	0
4	0	0	-1/2100	0	1/15750	0
5	0	-1/5040	0	0	0	1/332640

$$\int_{-1/2}^{+1/2} P_M(u) \cdot P_N^m(u) du$$

M \ N	0	1	2	3	4	5
0	0	0	2	0	2/5	0
1	0	0	0	1/2	0	1/12
2	0	0	0	0	1/15	0
3	0	0	0	-1/20	0	-1/840
4	0	0	0	0	-1/175	0
5	0	0	0	-1/840	0	-1/1680

APPENDIX B

Example 1

The IAEA-2D benchmark problem

One quadrant of the core is shown in Fig. B.1.

The group constants for the example are given below:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})	
1	1	1.5	0.01	0.0	0.02	Fuel 1
	2	0.4	0.08	0.135		
2	1	1.5	0.01	0.0	0.02	Fuel 2
	2	0.4	0.085	0.135		
3	1	1.5	0.01	0.0	0.02	Fuel 2 + rod
	2	0.4	0.13	0.135		
4	1	2.0	0.0	0.0	0.04	Reflector
	2	0.3	0.01	0.0		
5	1	2.0	0.0	0.0	0.04	Reflector + rod
	2	0.3	0.055	0.0		

$\chi_1 = 1.0$ $\chi_2 = 0.0$ $\nu = 2.43$

The axial buckling $B_z^2 = 0.8 \cdot 10^{-4}$ cm^{-2} for all regions and energy groups.

The external axial boundary conditions are $j_g^{in} = 0$, which are the same as zero albedo. Example 1 represents the midplane z = 190 cm of the 3D IAEA benchmark problem (Example 2).

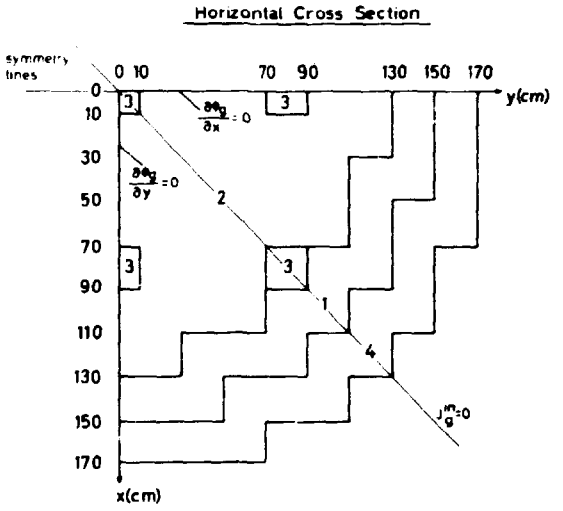


Fig. B.1. One quadrant of the core for the two- (and three-) dimensional IAEA benchmark.

Example 2

The IAEA-3D benchmark problem

One quadrant of the core is shown in Fig. B.1 and a vertical view of the problem is shown in Fig. B.2. The group constants are the same as for example 1.

Example 3

The Biblis-2D benchmark problem (Nakata and Martin, 1983).

The Biblis-2D benchmark is a checkerboard-loaded PWR core, and one eight of the core is shown in Fig. B.3.

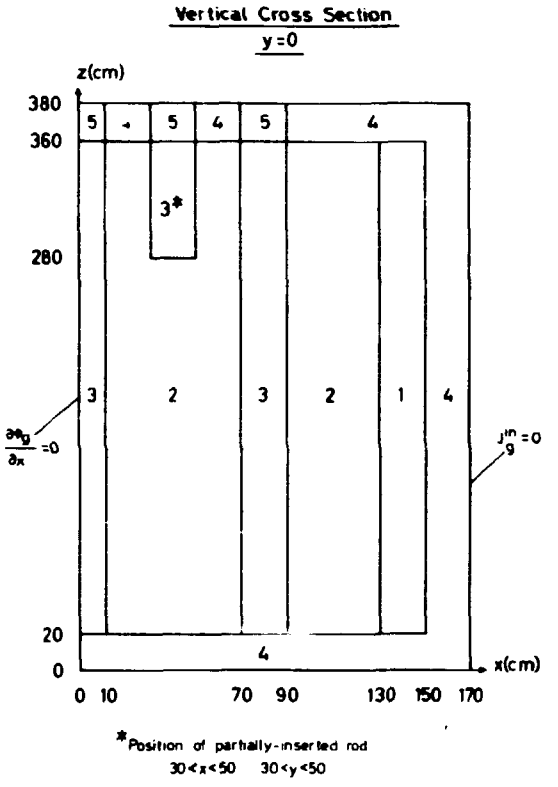


Fig. B.2. A vertical view of the fuel types for the three-dimensional IAEA benchmark.

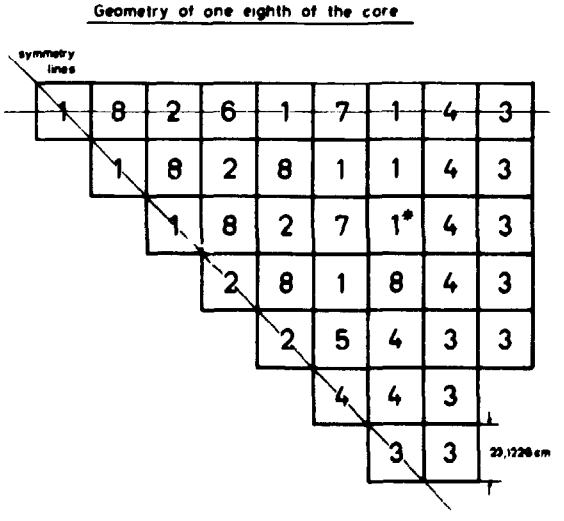


Fig. B.3. One eighth of the core for the two-dimensional Biblis benchmark.

The group constants are:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1	1	1.4360	0.0095042	0.0058708	0.017754
	2	0.3635	0.0750058	0.0960670	
2	1	1.4366	0.0096785	0.0061908	0.017621
	2	0.3636	0.0784360	0.1035800	
3	1	1.3200	0.0026562	0.0	0.023106
	2	0.2772	0.0715960	0.0	
4	1	1.4389	0.0103630	0.0074527	0.017101
	2	0.3638	0.0914080	0.1323600	
5	1	1.4381	0.0100030	0.0061908	0.017290
	2	0.3665	0.0848280	0.1035800	
6	1	1.4385	0.0101320	0.0064285	0.017192
	2	0.3665	0.0873140	0.1091100	
7	1	1.4389	0.0101650	0.0061908	0.017125
	2	0.3679	0.0880240	0.1035800	
8	1	1.4393	0.0102940	0.0064285	0.017027
	2	0.3680	0.0905100	0.1091100	

$$\chi_1 = 1.0 \qquad \chi_2 = 0.0 \qquad \nu = 2.47$$

The axial buckling $B_z^2 = 0.0\ cm^{-2}$ in all regions and groups.

The Biblis-2D benchmark are in 6 different configurations

Biblis 1: The reflector zone (region 3) is changed with some albedos:

$$\alpha_{11} = 0.4650 \qquad \alpha_{22} = 0.9807 \qquad \alpha_{12} = \alpha_{21} = 0.0$$

Biblis 2. A control rod has been inserted in region 1*

$$\Delta\Sigma_{a1} = 3.445 \cdot 10^{-3}\ cm^{-1} \quad \text{and} \quad \Delta\Sigma_{a2} = 2.030 \cdot 10^{-2}\ cm^{-1}$$

are added to the absorption cross sections. Albedo representation of the reflector.

Biblis 3/1. Rods out configuration (Biblis 1) but one zone with reflector cross sections (Fig. B.3). Outer albedos equal to zero.

Biblis 3/2. Same as 3/1, but with two reflector zones.

Biblis 4/1. Same as 3/1, but with control rod in.

Biblis 4/2. Same as 3/2, but with control rod in.

Example 4

A typical Westinghouse PWR (Larsen, 1983). One quadrant of the reactor is shown in Fig. B.4. The core height is 366 cm in the

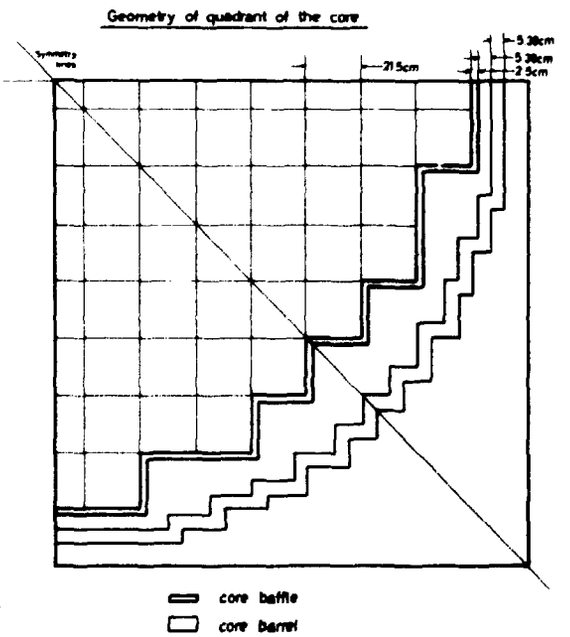


Fig. B.4. One quarter of the core for a reactor of typical Westinghouse PWR design.

three-dimensional example. The group constants are different for all nodes and they are calculated by the three-dimensional PWR transient code ANTI (Larsen, 1980) (Nielsen and Larsen, 1980) after a fully converged steady-state calculation. The physical and geometrical values for the albedo calculation are from Thorlaksen (1981). The group constants cannot be given here, since they are confidential.

Example 5

Infinite slab reactor model.

The infinite slab reactor is divided in 3 regions: two outer regions (1 and 3) 40 cm stick and a central region (2) 160 cm thick. The outer boundary conditions are $\phi = 0$.

The group constants are:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1,3	1	1.50	0.011	0.010	0.015
	2	0.50	0.180	0.200	
2	1	1.00	0.010	0.005	0.010
	2	0.50	0.080	0.099	

$\chi_1 = 1.00$ $\quad \chi_2 = 0.00$ $\quad \nu = 2.43$

$v_1 = 1.0 \cdot 10^7$ cm/s $\quad v_2 = 3.0 \cdot 10^5$ cm/s

Parameters for delayed neutrons (6 families):

Family	Delay fraction	Decay constant s^{-1}
1	0.00025	0.0124
2	0.00164	0.0305
3	0.00147	0.1110
4	0.00296	0.3010
5	0.00086	1.1400
6	0.00032	3.0100

Transient 1

Initial perturbation: Σ_{a2} in region 1 is linearly increased by 3% in 1.0 second.

The transient is calculated till time = 2.0 second.

Transient 2

Initial perturbation: Σ_{a2} in region 1 is linearly decreased by 1% in 1.0 second.

The transient is calculated till time = 4.0 second.

Transient 3

Initial perturbation: Σ_{a2} in region 1 is linearly decreased by 5% in 0.01 second.

The transient is calculated till time = 0.02 second.

Transient 4

Initial perturbation: Σ_{a2} in region 1 is linearly decreased by 5% in 0.01 second.

The neutron velocities are changed to

$v_1 = 1.0 \cdot 10^9$ cm/s and $v_2 = 3.0 \cdot 10^7$ cm/s

The transient is calculated till time = 0.005 second.

Example 6

The TWIGL two-dimensional seed-blanket reactor kinetics problem.

One quadrant of the core is shown in Fig. B.5.

The group constants for the example are given below:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1	1	1.4	0.01	0.007	0.01
	2	0.4	0.15	0.2	
2	1	1.4	0.1	0.007	0.01
	2	0.4	0.15	0.2	
3	1	1.3	0.008	0.003	0.01
	2	0.5	0.05	0.06	

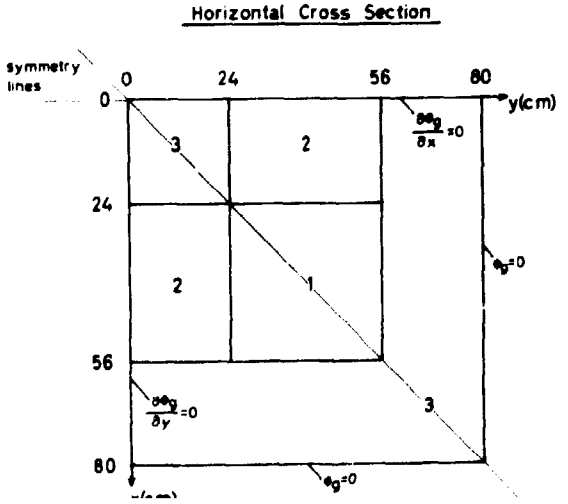


Fig. B.5. One quadrant of the core for the two-dimensional TWIGL benchmark.

$\chi_1 = 1.0$ $\chi_2 = 0.0$ $\nu = 2.43$

$v_1 = 1.0 \cdot 10^7$ cm/s $v_2 = 2.0 \cdot 10^5$ cm/s

The axial buckling $B_z^2 = 0.0$ cm^{-2} in all regions and groups.

Parameters for delayed neutrons (1 neutron family):

Family	Delay fraction	Decay constant (s^{-1})
1	0.0075	0.080

Transient 1

Step perturbation in region 1:

$$\Delta\Sigma_{a2} = -0.0035 \text{ cm}^{-1} \qquad t = 0$$

The transient is calculated for 0.5 seconds.

Transient 2

Ramp perturbation in region 1:

$$\Sigma_{a2}(t) = \begin{cases} 0.15 \cdot \left[1 - \frac{7}{300} \cdot \left(\frac{t}{0.2}\right)\right] \text{ cm}^{-1} & t < 0.2 \text{ seconds} \\ 0.1465 \text{ cm}^{-1} & t > 0.2 \text{ seconds} \end{cases}$$

The transient is calculated for 0.5 seconds.

Example 7

The LMWLWR transient problem.

One quadrant of the core is shown in Fig. B.6 and the initial and final rod positions are shown in Fig. B.7.

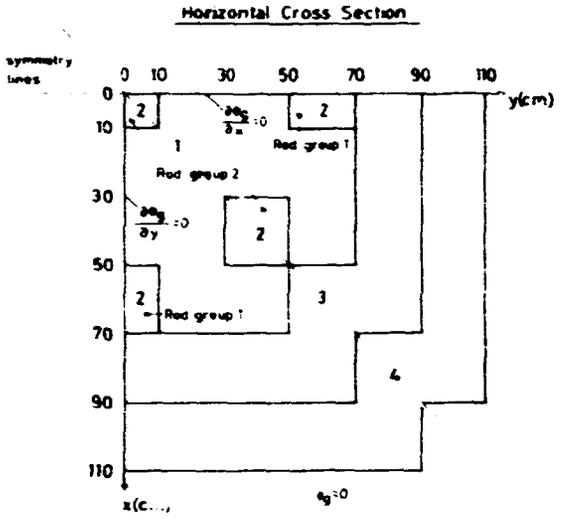


Fig. B.6. One quadrant of the core for the three-dimensional LMWLWR benchmark.

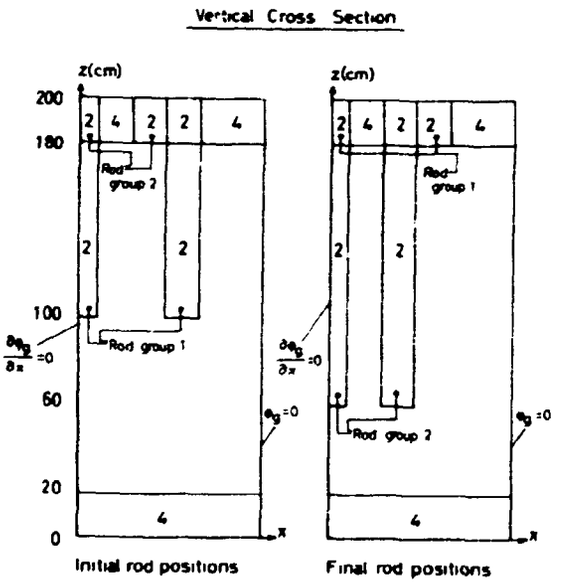


Fig. B.7. A vertical view of the core for the three-dimensional LMWLWR benchmark with the initial control rod pattern at left and the final one at right.

The group constants are:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1	1	1.423913	0.01040206	0.00647691	0.0175555
	2	0.356306	0.08766217	0.1127328	
2	1	1.423913	0.01095206	0.00647769	0.0175555
	2	0.356306	0.09146217	0.1127328	
3	1	1.425611	0.01099263	0.007503284	0.01717768
	2	0.350574	0.09925634	0.1378004	
4	1	1.634227	0.002660573	0.0	0.02759693
	2	0.264002	0.04936351	0.0	

$\chi_1 = 1.0$ $\chi_2 = 0.0$ $\nu = 2.5$

$v_1 = 1.25 \cdot 10^7$ cm/s $v_2 = 2.5 \cdot 10^5$ cm/s

Parameters for delayed neutrons (6 neutron families):

Family	Delay fraction	Decay constant (s^{-1})
1	0.000247	0.0127
2	0.0013845	0.0317
3	0.001222	0.115
4	0.0026455	0.311
5	0.000832	1.40
6	0.000169	3.87

Perturbation:

Rod group 1 removed at 3.0 cm/s for $0 < t < 26.666$ seconds.
Rod group 2 inserted at 3.0 cm/s for $7.5 < t < 7.5$ seconds.

The transient is calculated for 60 seconds.

Example 8

The LRABWR two-dimensional benchmark.

One quadrant of the core is shown in Fig. B.8.

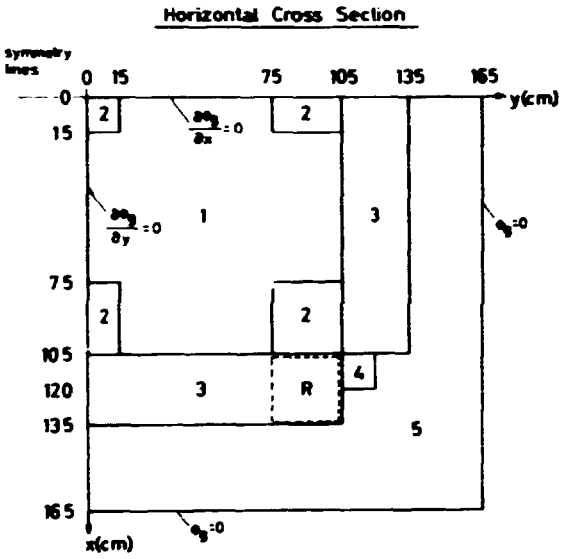


Fig. B.8. One quadrant of the core for the two- and three-dimensional LRABWR benchmark.

The group constants are:

Region	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1	1	1.255	0.008252	0.004602	0.02533
	2	0.211	0.1003	0.1091	
2	1	1.268	0.007181	0.004609	0.02767
	2	0.1902	0.07047	0.08675	
3,R	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.08344	0.1021	
4	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.073324	0.1021	
5	1	1.257	0.0006034	0.0	0.04754
	2	0.1592	0.01911	0.0	

$\chi_1 = 1.0$ $\chi_2 = 0.0$ $\nu = 2.43$

The axial buckling $B_z^2 = 1.0 \cdot 10^{-4}$ cm^{-2} in all regions and groups.

Parameters for delayed neutrons (2 neutron families):

Family	Delay fraction	Decay constant (s^{-1})
1	0.0054	0.0654
2	0.001087	1.35

Adiabatic heatup

$$\alpha[\Sigma_{f1}(\bar{r},t)\cdot\phi_1(\bar{r},t) + \Sigma_{f2}(\bar{r},t)\cdot\phi_2(\bar{r},t)] = \frac{\partial}{\partial t} T(\bar{r},t)$$

conversion factor: $\alpha = 3.83 \cdot 10^{-11}$ K $\cdot$ cm^3

Doppler feedback

$$\Sigma_{a1}(\bar{r},t) = \Sigma_{a1}(\bar{r},0)[1+\gamma(\sqrt{T(\bar{r},t)} - \sqrt{T_o})]$$

feedback constant: $\gamma = 2.034 \cdot 10^{-3}$ K$^{1/2}$

initial temperature: $T_o = 300$ K

Power

$$P(\bar{r},t) = \varepsilon[\Sigma_{f1}(\bar{r},t)\phi_1(\bar{r},t) + \Sigma_{f2}(\bar{r},t)\phi_2(\bar{r},t)]$$

energy conversion factor: $\varepsilon = 3.204 \cdot 10^{-11}$ W$\cdot$s/fission

The initial flux distribution shall be normalized such that the average power density

$$\bar{P} = \frac{\varepsilon}{V} \int_V (\Sigma_{f1}\Phi_1 + \Sigma_{f2}\Phi_2) dV = 1.0 \cdot 10^{-6} \text{ W cm}^{-3}$$

Perturbation

Control rod region (R) is given by

$$\Sigma_{a2}(t)) = \begin{cases} 0.08344 \cdot (1-0.0606184 \cdot t) \text{ cm}^{-1} & \text{for } t \leq 2.0 \text{ s} \\ 0.073324 \text{ cm}^{-1} & \text{for } t > 2.0 \text{ s} \end{cases}$$

The transient is calculated for 3.0 seconds.

Example 9

The LRABWR three-dimensional benchmark.

Example 9 is the three-dimensional case of example 8. A vertical view of the core is shown in Fig. B.9.

Perturbation

Control rod region (R) is removed with a velocity of 150 cm/s.

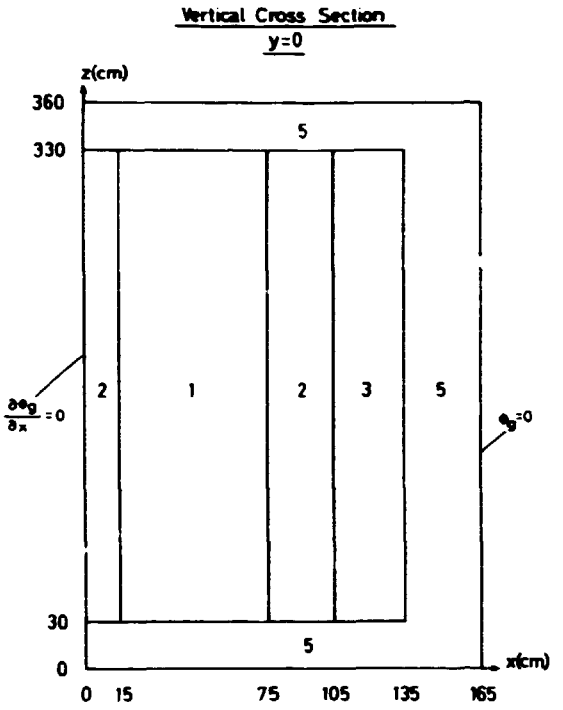


Fig. B.9. A vertical picture of the core for the three-dimensional LRABWR benchmark.

APPENDIX C

C.1 Solutions to the IAEA-2D benchmark

The reference power distribution is

	1	2	3	4	5	6	7	8
1	0.7456	1.3097	1.4537	1.2107	0.6100	0.9351	0.9343	0.7549
2		1.4351	1.4799	1.3149	1.0697	1.03[illegible]	0.9504	0.7358
3			1.4694	1.3451	1.1792	1.0705	0.9752	0.6921
4				1.1929	0.9670	0.9064	0.8461	
5					0.4706	0.6856	0.5972	
6						0.5849		

k_{eff} = 1.029585

The reference solution is calculated with a nodal program called MEDIUM-2 with a mesh length of 3 1/3 cm (Wagner, Finnemann, Koebke, and Winther, 1977).

The calculated power distribution for the 2D-benchmark with 30 cm nodes and orders 4,2

X/Y	1	2	3	4	5	6	7	8
1	0.7449	1.3126	1.4598	1.2132	0.6091	0.9351	0.9308	0.7477
2		1.4400	1.4840	1.3194	1.0712	1.0366	0.9459	0.7262
3			1.4718	1.3488	1.1828	1.0697	0.9576	0.6949
4				1.1966	0.9673	0.9036	0.8444	
5					0.4687	0.6792	0.6018	
6						0.5915		

The absolute errors in the solution above

X/Y	1	2	3	4	5	6	7	8
1	-0.0007	0.0029	0.0061	0.0025	-0.0009	0.0000	-0.0035	-0.0072
2		0.0049	0.0041	0.0045	0.0015	0.0005	-0.0045	-0.0096
3			0.0024	0.0037	0.0036	-0.0008	-0.0076	0.0028
4				0.0037	0.0003	-0.0028	-0.0017	
5					-0.0019	-0.0064	0.0046	
6						-0.0066		

C.2. Solutions to the IAEA-3D benchmark

The reference assembly power distribution is

X/Y	1	2	3	4	5	6	7	8
1	0.7290	1.2830	1.4230	1.1950	0.6100	0.9340	0.9580	0.7730
2		1.3980	1.4320	1.2920	1.0720	1.0550	0.9740	0.7530
3			1.3690	1.3110	1.1810	1.0880	0.9970	0.7070
4				1.1790	0.9720	0.9230	0.8640	
5					0.4750	0.6990	0.6630	
						0.5970		

$k_{eff.} = 1.02903$

The reference solution is calculated with a nodal program called IQSBOX with a node size of 10 × 10 × 10 cm^3 (Wagner, Finnemann, Koebke and Winther, 1977).

The calculated assembly power distribution with orders 4,2

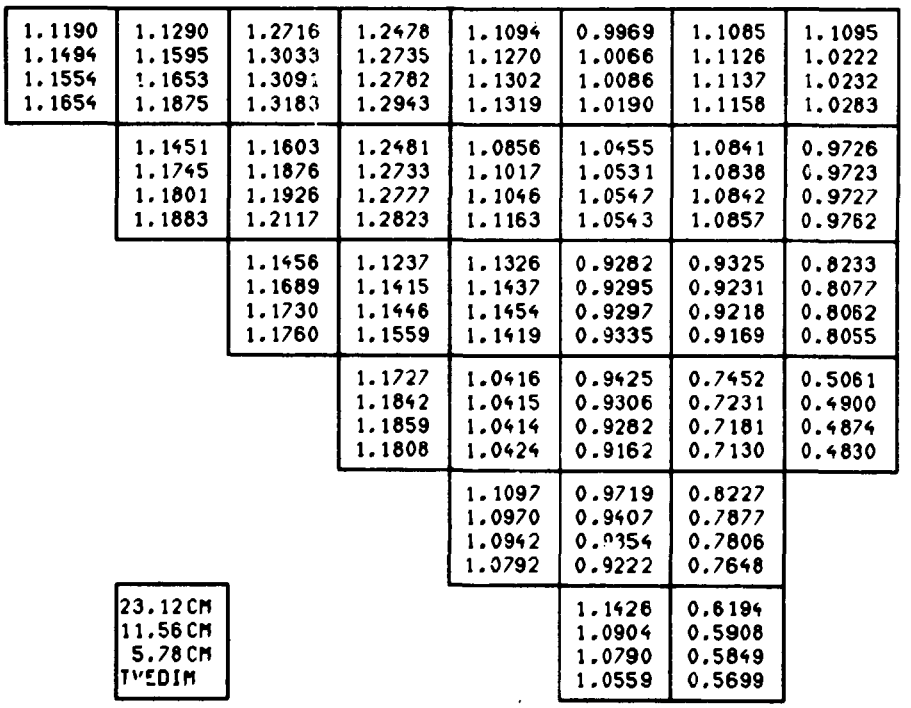
X/Y	1	2	3	4	5	6	7	8
1	0.7348	1.2922	1.4363	1.2001	0.6105	0.9488	0.9499	0.7650
2		1.4108	1.4441	1.2992	1.0722	0.0507	0.9648	0.7428
3			1.3815	1.3185	1.1832	1.0836	0.9864	0.7102
4				1.1829	0.9703	0.9162	0.8605	
5					0.4734	0.6903	0.6133	
6						0.6024		

The absolute errors in the above solution

X/Y	1	2	3	4	5	6	7	8
1	0.0058	0.0092	0.0133	0.0051	0.0005	-0.0052	-0.0081	-0.0080
2		0.0128	0.0121	0.0072	0.0002	-0.0043	-0.0092	-0.0102
3			0.0125	0.0075	0.0022	-0.0044	-0.0106	0.0032
4				0.0039	-0.0017	-0.0068	-0.0035	
5					-0.0016	-0.0087	0.0053	
6						0.0054		

C.3. Solutions to the Biblis benchmark

BIBLIS-1



BIBLIS-2

1.6130	1.6061	1.7403	1.5945	1.2569	0.9245	0.8437	0.7146
1.6462	1.6393	1.7744	1.6213	1.2753	0.9343	0.8470	0.7164
1.6534	1.6462	1.7812	1.6266	1.2787	0.9361	0.8479	0.7170
1.6685	1.6685	1.7952	1.6487	1.2818	0.9457	0.8481	0.7190
	1.6101	1.5709	1.5790	1.2148	0.9302	0.7350	0.6220
	1.6418	1.5998	1.6054	1.2309	0.9377	0.7370	0.6219
	1.6484	1.6057	1.6103	1.2338	0.9390	0.7375	0.6220
	1.6611	1.6327	1.6176	1.2478	0.9381	0.7361	0.6224
		1.4998	1.3813	1.2429	0.7799	0.4180	0.4408
		1.5243	1.3992	1.2532	0.7822	0.4137	0.4294
		1.5289	1.4025	1.2549	0.7824	0.4129	0.4282
		1.5343	1.4175	1.2512	0.7836	0.4130	0.4246
			1.3777	1.1292	0.8756	0.5376	0.3066
			1.3882	1.1273	0.8625	0.5180	0.2933
			1.3897	1.1269	0.8597	0.5135	0.2913
			1.3844	1.1272	0.8459	0.5055	0.2867
				1.1761	0.9806	0.7826	
				1.1599	0.9455	0.7432	
				1.1563	0.9392	0.7354	
				1.1380	0.9221	0.7156	
23.12CM					1.1791	0.6349	
11.56CM					1.1174	0.5992	
5.78CM					1.1043	0.5922	
TVEDIM					1.0751	0.5736	

BIBLIS-3/1

1.0855	1.0962	1.2378	1.2189	1.0885	0.9827	1.0956	1.0091
1.0938	1.1048	1.2463	1.2248	1.0921	0.9823	1.0894	1.0052
1.0945	1.1055	1.2468	1.2243	1.0905	0.9815	1.0911	1.0099
1.0946	1.0941	1.2464	1.2167	1.0942	0.9779	1.0993	1.0168
	1.1129	1.1308	1.2212	1.0675	1.0333	1.0745	0.9654
	1.1206	1.1379	1.2280	1.0716	1.0326	1.0676	0.9624
	1.1211	1.1381	1.2271	1.0698	1.0313	1.0686	0.9663
	1.1212	1.1274	1.2286	1.0632	1.0366	1.0754	0.9727
		1.1204	1.1049	1.1209	0.9255	0.9346	0.8235
		1.1264	1.1100	1.1245	0.9253	0.9286	0.8193
		1.1258	1.1084	1.1219	0.9235	0.9285	0.8214
		1.1269	1.1002	1.1254	0.9201	0.9357	0.8261
			1.1620	1.0423	0.9543	0.7689	0.5412
			1.1661	1.0425	0.9513	0.7637	0.5427
			1.1635	1.0401	0.9495	0.7636	0.5438
			1.1660	1.0344	0.9548	0.7631	0.5457
				1.1269	1.0017	0.8728	
				1.1242	0.9917	0.8711	
				1.1225	0.9917	0.8725	
				1.1265	0.9920	0.8743	
23.12CM					1.2080	0.6764	
11.56CM					1.1943	0.6794	
5.78CM					1.1967	0.6825	
TWODIM					1.2009	0.6838	

BIBLIS-3/2

1.0875 1.0950 1.0949 1.0744	1.0980 1.1060 1.1057 1.0975	1.2395 1.2473 1.2466 1.2267	1.2200 1.2247 1.2237 1.2165	1.0889 1.0907 1.0899 1.0784	0.9826 0.9816 0.9816 0.9874	1.0953 1.0917 1.0923 1.0965	1.0089 1.0098 1.0119 1.0256
	1.1147 1.1217 1.1213 1.1010	1.1323 1.1384 1.1378 1.1309	1.2223 1.2274 1.2264 1.2100	1.0679 1.0699 1.0691 1.0691	1.0331 1.0315 1.0312 1.0278	1.0741 1.0692 1.0696 1.0745	0.9652 0.9662 0.9681 0.9811
		1.1215 1.1262 1.1252 1.1076	1.1055 1.1085 1.1074 1.1042	1.1210 1.1220 1.1209 1.1109	0.9253 0.9235 0.9232 0.9285	0.9341 0.9288 0.9291 0.9312	0.8231 0.8212 0.8227 0.8331
			1.1622 1.1635 1.1622 1.1501	1.0420 1.0400 1.0391 1.0420	0.9537 0.9496 0.9491 0.9464	0.7683 0.7633 0.7637 0.7720	0.5409 0.5433 0.5444 0.5520
				1.1262 1.1225 1.1217 1.1189	1.0007 0.9915 0.9914 0.9974	0.8719 0.8718 0.8727 0.8814	
23.12CM 11.56CM 5.78CM TVEDIM					1.2066 1.1963 1.1968 1.2065	0.6755 0.6811 0.6827 0.6918	

BIBLIS-4/1

1.5752 1.5818 1.5823 1.5859	1.5694 1.5764 1.5768 1.5653	1.7032 1.7100 1.7101 1.7147	1.5641 1.5676 1.5673 1.5603	1.2364 1.2387 1.2383 1.2430	0.9124 0.9131 0.9130 0.9083	0.8347 0.8333 0.8334 0.8370	0.7069 0.7093 0.7104 0.7131
	1.5743 1.5802 1.5804 1.5854	1.5388 1.5437 1.5437 1.5339	1.5509 1.5552 1.5549 1.5593	1.1972 1.1990 1.1986 1.1917	0.9201 0.9197 0.9194 0.9232	0.7290 0.7286 0.7287 0.7324	0.6182 0.6209 0.6218 0.6246
		1.4730 1.4765 1.4761 1.4817	1.3621 1.3640 1.3634 1.3554	1.2320 1.2323 1.2316 1.2360	0.7784 0.7785 0.7783 0.7755	0.4210 0.4200 0.4200 0.4182	0.4446 0.4455 0.4462 0.4490
			1.3674 1.3675 1.3666 1.3715	1.1308 1.1278 1.1272 1.1215	0.8877 0.8834 0.8828 0.8872	0.5599 0.5564 0.5566 0.5561	0.3428 0.3448 0.3454 0.3460
				1.1945 1.1892 1.1884 1.1927	1.0108 0.9993 0.9991 0.9966	0.8246 0.8214 0.8220 0.8228	
23.12CM 11.56CM 5.78CM TWODIM					1.2443 1.2290 1.2293 1.2322	0.6892 0.6913 0.6928 0.6932	

BIBLIS-4/2

1.5724	1.5666	1.7005	1.5620	1.2352	0.9120	0.8349	0.7075
1.5798	1.5745	1.7082	1.5663	1.2380	0.9130	0.8336	0.7101
1.5806	1.5752	1.7086	1.5662	1.2377	0.9129	0.8337	0.7112
1.5638	1.5757	1.6926	1.5647	1.2273	0.9168	0.8335	0.7176
	1.5716	1.5365	1.5490	1.1962	0.9198	0.7293	0.6189
	1.5784	1.5422	1.5540	1.1984	0.9196	0.7290	0.6216
	1.5789	1.5424	1.5538	1.1981	0.9193	0.7290	0.6225
	1.5620	1.5431	1.5403	1.2004	0.9143	0.7282	0.6279
		1.4711	1.3609	1.2316	0.7785	0.4213	0.4453
		1.4753	1.3632	1.2320	0.7786	0.4203	0.4461
		1.4751	1.3628	1.2313	0.7784	0.4203	0.4468
		1.4602	1.3643	1.2223	0.7806	0.4237	0.4498
			1.3670	1.1313	0.8887	0.5609	0.3435
			1.3672	1.1280	0.8840	0.5570	0.3454
			1.3663	1.1273	0.8832	0.5570	0.3459
			1.3562	1.1323	0.8807	0.5616	0.3501
				1.1961	1.0129	0.8267	
				1.1900	1.0004	0.8226	
				1.1890	0.9999	0.8229	
				1.1877	1.0069	0.8314	
	23.12CM				1.2476	0.6912	
	11.56CM				1.2308	0.6925	
	5.78CM				1.2306	0.6937	
	TVEDIM				1.2421	0.7036	

BIBLIS with new albedos

1.0814	1.0921	1.2334	1.2150	1.0859	0.9820	1.0974	1.0136
1.1084	1.1193	1.2617	1.2379	1.1016	0.9906	1.1010	1.0159
1.1136	1.1243	1.2667	1.2420	1.1044	0.9924	1.1019	1.0167
1.0744	1.0975	1.2267	1.2165	1.0784	0.9874	1.0965	1.0256
	1.1089	1.1269	1.2175	1.0650	1.0325	1.0766	0.9705
	1.1349	1.1512	1.2400	1.0795	1.0393	1.0760	0.9699
	1.1397	1.1556	1.2438	1.0820	1.0407	1.0763	0.9702
	1.1010	1.1309	1.2100	1.0691	1.0278	1.0745	0.9811
		1.1168	1.1018	1.1186	0.9245	0.9364	0.8321
		1.1376	1.1178	1.1287	0.9257	0.9273	0.8177
		1.1412	1.1206	1.1303	0.9259	0.9260	0.8161
		1.1076	1.1042	1.1109	0.9285	0.9312	0.8331
			1.1595	1.0411	0.9543	0.7650	0.5403
			1.1701	1.0413	0.9432	0.7449	0.5246
			1.1716	1.0413	0.9411	0.7402	0.5218
			1.1501	1.0420	0.9464	0.7720	0.5520
				1.1279	1.0058	0.8635	
				1.1172	0.9780	0.8335	
				1.1149	0.9735	0.8279	
				1.1189	0.9974	0.8814	
	23.12CM				1.2200	0.6921	
	11.56CM				1.1745	0.6667	
	5.78CM				1.1651	0.6619	
	TVEDIM				1.2065	0.6918	

C.4. Solutions to the typical Westinghouse PWR

The reference assembly power distribution is

X/Y	1	2	3	4	5	6	7	8
1	1.19	1.11	1.22	1.10	1.18	1.14	0.99	0.84
2		1.22	1.18	1.20	1.11	1.09	1.00	0.65
3			1.21	1.13	1.09	0.94	0.95	
4				1.10	0.94	0.92	0.66	
5					0.82	0.66		

and the vertical distribution

NODE	POWER
20	0.1916
19	0.3950
18	0.5718
17	0.7245
16	0.8575
15	0.9747
14	1.0790
13	1.1716
12	1.2529
11	1.3222
10	1.3775
9	1.4159
8	1.4330
7	1.4229
6	1.3783
5	1.2907
4	1.1512
3	0.9515
2	0.6857
1	0.3524

k_{eff} = 0.99000

The reference solution is obtained with the ANTI-code (Larsen, 1983)

The calculated assembly powers for example 4 by use of the reflector parameters calculated by Thorlaksen (1981). The node size = 21.5 × 21.5 × 18.3 cm^3.

X/Y	1	2	3	4	5	6	7	8
1	1.03	1.00	1.09	1.06	1.17	1.22	1.11	0.94
	1.00	0.98	1.07	1.05	1.16	1.21	1.12	0.97
	1.02	0.99	1.09	1.06	1.17	1.21	1.10	0.95
	1.19	1.11	1.22	1.10	1.18	1.14	0.99	0.84
2		1.07	1.09	1.13	1.13	1.15	1.06	0.69
		1.05	1.07	1.11	1.12	1.15	1.07	0.71
		1.07	1.09	1.13	1.13	1.15	1.06	0.71
		1.22	1.18	1.20	1.11	1.09	1.00	0.65
3			1.12	1.11	1.10	1.01	0.94	
			1.11	1.09	1.09	1.02	0.96	
			1.12	1.10	1.10	1.01	0.95	
			1.21	1.13	1.09	0.94	0.95	
4				1.09	0.99	0.94	0.63	
				1.08	0.98	0.94	0.65	
				1.08	0.98	0.94	0.65	
				1.10	0.94	0.92	0.66	
5	NEM with α_T^D				0.88	0.66		
	NEM with α_T				0.89	0.68		
	QUANDRY				0.88	0.67		
	ANTI (reference)				0.82	0.66		

The eigenvalues are

NEM α_T^D	0.99399
NEM α_T	0.99443
QUANDRY	0.99417
ANTI	0.99000

The calculated assembly powers for example 4 with new albedo representation.

X/Y	1	2	3	4	5	6	7	8
1	0.99	0.97	1.06	1.04	1.16	1.22	1.10	0.96
	0.96	0.94	1.03	1.02	1.14	1.20	1.08	0.94
	1.19	1.11	1.22	1.10	1.18	1.14	0.99	0.84
2		1.04	1.06	1.11	1.13	1.15	1.08	0.71
		1.01	1.03	1.08	1.11	1.14	1.08	0.75
		1.22	1.18	1.20	1.11	1.09	1.00	0.65
3			1.07	1.09	1.10	1.03	0.97	
			1.05	1.07	1.08	1.02	1.01	
			1.21	1.13	1.09	0.94	0.95	
4				1.08	0.99	0.96	0.66	
				1.07	0.99	0.98	0.71	
				1.10	0.94	0.92	0.66	
5	NEM with new albedos				0.90	0.68		
	NEM with cross sections				0.92	0.77		
	ANTI (reference)				0.82	0.66		

The eigenvalues are

NEM	Albedo	0.99461
NEM	cross sections	0.99509
ANTI		0.99000

C.5 Solutions to the infinite slab model

Infinite slab reactor model. +3% thermal absorption cross section in region 1 in 1.0 second.

Time Sec.	Relative regional power Region 1	Region 2	Region 3	Total power	Total power (reference)
0.0	1.000	1.000	1.000	1.000	1.000
0.1	0.859	0.929	0.989	0.926	0.930
0.2	0.747	0.875	0.982	0.869	0.873
0.3	0.656	0.828	0.969	0.819	
0.4	0.586	0.792	0.962	0.782	
0.5	0.523	0.760	0.955	0.748	0.760
0.6	0.477	0.735	0.947	0.722	
0.7	0.433	0.712	0.941	0.698	
0.8	0.397	0.692	0.936	0.678	
0.9	0.362	0.672	0.930	0.658	
1.0	0.339	0.660	0.925	0.644	0.659
1.5	0.316	0.639	0.909	0.624	0.643
2.0	0.298	0.619	0.891	0.605	0.631

Infinite slab reactor model. -1% thermal absorption cross section in region 1 in 1.0 second.

Time Sec.	Relative regional power Region 1	Region 2	Region 3	Total power	Total power (reference)
0.0	1.000	1.000	1.000	1.000	1.000
0.1	1.057	1.026	1.004	1.029	1.028
0.2	1.122	1.055	1.004	1.059	1.062
0.3	1.195	1.089	1.008	1.096	
0.4	1.276	1.127	1.011	1.136	
0.5	1.380	1.176	1.016	1.188	1.205
0.6	1.499	1.232	1.022	1.248	
0.7	1.651	1.304	1.031	1.325	
0.8	1.828	1.390	1.042	1.415	
0.9	2.055	1.500	1.057	1.531	
1.0	2.347	1.641	1.076	1.680	1.740
1.5	2.679	1.811	1.105	1.856	1.959
2.0	2.971	1.962	1.133	2.012	2.165
2.5	3.302	2.133	1.165	2.189	
3.0	3.558	2.266	1.193	2.327	2.605
3.5	3.880	2.436	1.230	2.502	
4.0	4.179	2.591	1.260	2.662	3.107

Infinite slab reactor model. -5% thermal absorption cross section in region 1 in 0.01 seconds.

Time Sec.	Relative regional power Region 1	Region 2	Region 3	Total power	Total power (reference)
0.000	1.000	1.000	1.000	1.000	1.000
0.001	1.064	1.015	1.002	1.025	1.022
0.002	1.216	1.066	1.003	1.090	
0.003	1.470	1.160	1.005	1.203	
0.004	1.875	1.313	1.011	1.385	
0.005	2.507	1.551	1.019	1.669	1.659
0.008	9.234	3.966	1.106	4.636	
0.010	35.19	12.68	1.342	15.79	15.65
0.012	163.1	56.34	2.403	71.05	70.18
0.015	1582	542.3	15.97	685.0	680.2
0.018	$1.528 \cdot 10^4$	5239	136.7	6620	6611
0.020	$6.934 \cdot 10^4$	$2.375 \cdot 10^4$	619.6	$3.000 \cdot 10^4$	$3.011 \cdot 10^4$

Infinite slab reactor model. -5% thermal absorption cross section in region 1 in 0.01 seconds. Hot case.

Time Sec.	Relative regional power Region 1	Region 2	Region 3	Total power	Total power (reference)
0.000	1.000	1.000	1.000	1.000	1.000
0.001	1.354	1.168	1.025	1.180	1.178
0.002	2.109	1.527	1.073	1.562	1.558
0.003	4.573	2.694	1.226	2.809	2.797
0.004	40.57	19.44	3.233	20.81	20.72
0.005	$3.175 \cdot 10^5$	$1.394 \cdot 10^5$	$1.275 \cdot 10^4$	$1.537 \cdot 10^5$	$1.537 \cdot 10^5$

C.6 Solutions to the TWIGL example

The steady-state power solution

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.258	1.293	1.243	2.373	2.176	1.967	1.691	0.647	0.422	0.139
2		1.321	1.259	2.380	2.161	1.941	1.663	0.635	0.414	0.136
3			1.198	2.350	2.123	1.883	1.602	0.609	0.396	0.130
4				2.187	2.033	1.779	1.500	0.568	0.368	0.121
5					1.870	1.614	1.350	0.509	0.329	0.108
6						1.380	1.148	0.432	0.279	0.091
7							0.948	0.343	0.220	0.072
8								0.260	0.157	0.051
9									0.093	0.030
10										0.010

k_{eff} = 0.91318

Normalized power distribution for TWIGL with ramp-perturbation to different times

T = 0.0 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.258	1.293	1.243	2.373	2.176	1.967	1.691	0.647	0.422	0.139
2		1.321	1.259	2.380	2.161	1.941	1.663	0.635	0.414	0.136
3			1.198	2.350	2.123	1.883	1.602	0.609	0.396	0.130
4				2.187	2.033	1.779	1.500	0.568	0.368	0.121
5					1.870	1.614	1.350	0.509	0.329	0.108
6						1.380	1.148	0.432	0.279	0.091
7							0.948	0.343	0.220	0.072
8								0.260	0.157	0.051
9									0.093	0.030
10										0.010

T = 0.05 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.251	1.286	1.237	2.362	2.167	1.959	1.685	0.645	0.420	0.138
2		1.314	1.253	2.371	2.155	1.936	1.658	0.663	0.412	0.135
3			1.193	2.346	2.121	1.883	1.602	0.609	0.395	0.130
4				2.194	2.044	1.790	1.509	0.569	0.368	0.121
5					1.885	1.629	1.362	0.511	0.330	0.108
6						1.394	1.160	0.434	0.280	0.092
7							0.957	0.346	0.221	0.072
8								0.261	0.158	0.052
9									0.093	0.031
10										0.010

T = 0.10 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.243	1.278	1.230	2.351	2.157	1.951	1.678	0.643	0.419	0.138
2		1.307	1.247	2.361	2.147	1.930	1.654	0.632	0.411	0.135
3			1.189	2.341	2.119	1.882	1.601	0.609	0.395	0.129
4				2.201	2.054	1.801	1.518	0.571	0.369	0.121
5					1.899	1.643	1.374	0.514	0.331	0.108
6						1.409	1.172	0.437	0.282	0.092
7							0.968	0.348	0.222	0.073
8								0.263	0.158	0.052
9									0.094	0.031
10										0.010

T = 0.15 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.235	1.271	1.223	2.338	2.147	1.942	1.671	0.640	0.418	0.137
2		1.299	1.241	2.351	2.139	1.924	1.649	0.630	0.410	0.135
3			1.184	2.336	2.117	1.882	1.601	0.608	0.395	0.129
4				2.208	2.065	1.812	1.527	0.572	0.369	0.121
5					1.914	1.659	1.387	0.516	0.332	0.109
6						1.424	1.185	0.440	0.283	0.092
7							0.978	0.351	0.224	0.073
8								0.265	0.159	0.052
9									0.094	0.031
10										0.010

T = 0.20 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.227	1.263	1.216	2.326	2.137	1.934	1.665	0.638	0.416	0.137
2		1.292	1.234	2.341	2.131	1.918	1.644	0.628	0.409	0.134
3			1.180	2.330	2.115	1.881	1.601	0.608	0.395	0.129
4				2.215	2.075	1.823	1.537	0.574	0.370	0.121
5					1.930	1.674	1.400	0.519	0.334	0.109
6						1.439	1.198	0.443	0.284	0.093
7							0.989	0.353	0.225	0.074
8								0.267	0.160	0.052
9									0.095	0.031
10										0.010

T = 0.25 - 0.50 seconds

X/Y	1	2	3	4	5	6	7	8	9	10
1	1.228	1.263	1.217	2.326	2.136	1.934	1.664	0.638	0.416	0.137
2		1.293	1.235	2.341	2.131	1.917	1.648	0.628	0.409	0.134
3			1.180	2.330	2.115	1.881	1.601	0.608	0.395	0.129
4				2.214	2.075	1.822	1.537	0.574	0.370	0.121
5					1.929	1.674	1.400	0.519	0.334	0.109
6						1.439	1.198	0.443	0.285	0.093
7							0.989	0.353	0.225	0.074
8								0.267	0.160	0.053
9									0.095	0.031
10										0.010

C.7. Solutions to the LMWLWR transient problem

The steady-state assembly power distribution obtained by NEM of orders 4,2.

X/Y	1	2	3	4	5
1	1.560	1.476	1.283	0.982	0.642
2		1.417	1.244	0.964	0.626
3			1.126	0.873	0.554
4				0.763	0.380

k_{eff} = 0.99953

C.8. Solutions to the LRABWR two-dimensional benchmark

The steady-state power distribution for the rod-in situation obtained by NEM of orders 4,2 and a node size of 15 × 15 cm^2.

X/Y	1	2	3	4	5	6	7	8	9
1	0.6360	0.4538	0.4230	0.5208	0.8012	1.4082	1.6848	1.4925	0.9244
2	0.4538	0.4107	0.4157	0.4986	0.6795	0.9492	1.1591	1.2876	0.8663
3	0.4230	0.4157	0.4316	0.4985	0.6238	0.7866	0.9688	1.1727	0.8222
4	0.5208	0.4986	0.4985	0.5570	0.6806	0.8424	1.0185	1.2148	0.8440
5	0.8012	0.6795	0.6238	0.6806	0.8634	1.1454	1.3286	1.4092	0.9189
6	1.4082	0.9492	0.7866	0.8424	1.1454	1.8445	2.0382	1.6592	0.9520
7	1.6848	1.1591	0.9688	1.0185	1.3286	2.0382	2.1417	1.5931	0.8263
8	1.4925	1.2876	1.1727	1.2148	1.4092	1.6592	1.5931	1.2986	
9	0.9244	0.8663	0.8222	0.8440	0.9189	0.9520	0.8263		

k_{eff} = 0.99619

The steady-state power distribution for the rod-out situation in cold state (no hydraulic feedback)

X/Y	1	2	3	4	5	6	7	8	9
1	0.2052	0.1533	0.1543	0.1997	0.3102	1.5443	0.6543	0.5854	0.3670
2	0.1589	0.1532	0.1691	0.2161	0.3005	1.4165	0.5107	0.5765	0.3914
3	0.1733	0.1823	0.2114	0.2702	0.3579	0.4594	0.5644	0.6774	0.4729
4	0.2447	0.2545	0.2959	0.3873	0.5274	0.6822	0.8173	0.9448	0.6417
5	0.4078	0.3833	0.4301	0.5845	0.8666	1.2260	1.4088	1.4262	0.8915
6	0.7473	0.5653	0.6053	0.8530	1.4034	2.4385	2.6806	2.0683	1.1233
7	0.9244	0.7274	0.8118	1.1736	1.9685	3.4210	3.5900	2.4560	1.1556
8	0.8452	0.8500	1.0381	1.5202	2.4758	4.0898	4.0068	2.5301	
9	0.5373	0.5891	0.7515	1.1065	1.7659	2.7687	2.5553		

k_{eff} = 1.01508

Normalized power distribution to time = 0.4 second. Average power density = $1.38 \cdot 10^{-6}$ W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.564	0.404	0.380	0.472	0.727	1.283	1.536	1.359	0.843
2	0.405	0.370	0.377	0.457	0.626	0.872	1.066	1.189	0.799
3	0.384	0.380	0.399	0.467	0.588	0.744	0.917	1.109	0.778
4	0.481	0.464	0.471	0.537	0.665	0.829	1.001	1.190	0.825
5	0.746	0.641	0.600	0.673	0.875	1.171	1.357	1.435	0.931
6	1.322	0.899	0.766	0.851	1.191	1.949	2.156	1.743	0.995
7	1.587	1.103	0.953	1.046	1.419	2.231	2.348	1.730	0.889
8	1.408	1.235	1.161	1.261	1.545	1.933	1.872	1.469	0.000
9	0.875	0.832	0.816	0.879	1.016	1.139	0.996	0.000	0.000

Time = 0.8 seconds. Average power density = $2.94 \cdot 10^{-6}$ W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.479	0.345	0.328	0.409	0.632	1.114	1.335	1.183	0.736
2	0.348	0.320	0.330	0.403	0.553	0.771	0.942	1.054	0.710
3	0.337	0.337	0.359	0.426	0.541	0.686	0.845	1.022	0.716
4	0.431	0.421	0.437	0.510	0.643	0.808	0.975	1.154	0.797
5	0.678	0.591	0.570	0.663	0.886	1.201	1.391	1.459	0.940
6	1.209	0.835	0.740	0.862	1.250	2.075	2.294	1.837	1.041
7	1.459	1.033	0.933	1.082	1.540	2.480	2.610	1.894	0.957
8	1.300	1.165	1.146	1.322	1.728	2.312	2.245	1.673	0.000
9	0.810	0.789	0.811	0.931	1.158	1.409	1.252	0.000	0.000

Time = 1.2 seconds. Average power density = $2.34 \cdot 10^{-4}$ W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.372	0.271	0.262	0.331	0.511	0.899	1.079	0.959	0.597
2	0.276	0.257	0.271	0.335	0.461	0.641	0.785	0.880	0.594
3	0.278	0.282	0.308	0.373	0.480	0.610	0.752	0.907	0.635
4	0.367	0.365	0.392	0.474	0.613	0.777	0.937	1.100	0.756
5	0.589	0.524	0.529	0.647	0.896	1.233	1.425	1.478	0.942
6	1.059	0.751	0.705	0.873	1.323	2.232	2.464	1.950	1.088
7	1.287	0.939	0.904	1.128	1.697	2.804	2.950	2.100	1.034
8	1.155	1.069	1.123	1.401	1.974	2.834	2.761	1.945	0.000
9	0.724	0.728	0.799	0.996	1.348	1.790	1.614	0.000	0.000

Time = 1.4 seconds. Average power density = 9.11 W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.328	0.241	0.235	0.297	0.460	0.807	0.968	0.862	0.537
2	0.246	0.231	0.246	0.306	0.421	0.585	0.716	0.803	0.542
3	0.252	0.258	0.285	0.349	0.452	0.576	0.708	0.853	0.596
4	0.339	0.340	0.371	0.456	0.597	0.759	0.914	1.069	0.731
5	0.547	0.493	0.509	0 637	0.896	1.240	1.431	1.475	0.935
6	0.986	0.709	0.685	0.875	1.351	2.291	2.527	1.988	1.102
7	1.202	0.892	0.887	1.145	1.767	2.952	3.103	2.189	1.065
8	1.083	1.021	1.108	1.435	2.092	3.106	3.030	2.076	0.000
9	0.680	0.697	0.790	1.024	1.442	1.994	1.807	0.000	0.000

Time = 1.6 seconds. Average power density = 154 W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.374	0.270	0.257	0.319	0.485	0.846	1.010	0.897	0.560
2	0.275	0.255	0.265	0.323	0.439	0.605	0.737	0.826	0.559
3	0.274	0.277	0.301	0.361	0.460	0.580	0.710	0.856	0.601
4	0.357	0.355	0.381	0.459	0.591	0.744	0.892	1.046	0.720
5	0.566	0.505	0.514	0.631	0.872	1.196	1.375	1.421	0.907
6	1.010	0.721	0.685	0.859	1.309	2.208	2.428	1.909	1.064
7	1.226	0.902	0.884	1.125	1.726	2.884	3.024	2.123	1.035
8	1.102	1.031	1.107	1.420	2.076	3.146	3.068	2.060	0.000
9	0.694	0.707	0.794	1.022	1.453	2.065	1.879	0.000	0.000

Time = 1.8 seconds. Average power density = 341 W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.329	0.239	0.229	0.285	0.433	0.754	0.901	0.801	0.501
2	0.244	0.228	0.239	0.293	0.398	0.548	0.667	0.748	0.507
3	0.247	0.251	0.276	0.335	0.429	0.542	0.664	0.799	0.560
4	0.326	0.328	0.358	0.439	0.571	0.722	0.864	1.009	0.693
5	0.522	0.472	0.491	0.617	0.866	1.195	1.372	1.410	0.896
6	0.936	0.677	0.662	0.855	1.329	2.259	2.481	1.939	1.074
7	1.139	0.852	0.864	1.138	1.793	3.034	3.179	2.210	1.065
8	1.028	0.980	1.089	1.452	2.200	3.455	3.372	2.202	0.000
9	0.650	0.674	0.784	1.052	1.557	2.307	2.112	0.000	0.000

Time = 2.0 seconds. Average power density = 1093 W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.297	0.216	0.208	0.259	0.393	0.683	0.815	0.725	0.454
2	0.222	0.208	0.219	0.269	0.366	0.502	0.610	0.686	0.466
3	0.227	0.232	0.257	0.315	0.404	0.510	0.623	0.749	0.526
4	0.303	0.306	0.339	0.421	0.551	0.698	0.834	0.971	0.666
5	0.486	0.444	0.470	0.602	0.855	1.185	1.358	1.388	0.879
6	0.873	0.639	0.641	0.848	1.338	2.286	2.506	1.948	1.073
7	1.066	0.809	0.843	1.144	1.843	3.153	3.301	2.273	1.085
8	0.966	0.935	1.071	1.474	2.306	3.745	3.660	2.326	0.000
9	0.613	0.647	0.775	1.076	1.652	2.547	2.344	0.000	0.000

Time = 3.0 seconds. Average power density = 74.1 W/cm^3

X/Y	1	2	3	4	5	6	7	8	9
1	0.331	0.240	0.228	0.281	0.423	0.734	0.875	0.777	0.487
2	0.245	0.228	0.237	0.288	0.389	0.533	0.647	0.726	0.493
3	0.245	0.249	0.273	0.330	0.420	0.528	0.644	0.775	0.544
4	0.321	0.323	0.352	0.431	0.558	0.704	0.840	0.980	0.674
5	0.510	0.462	0.482	0.605	0.850	1.171	1.342	1.376	0.875
6	0.912	0.661	0.650	0.844	1.315	2.237	2.452	1.910	1.056
7	1.109	0.832	0.850	1.131	1.800	3.066	3.209	2.214	1.061
8	1.002	0.959	1.076	1.454	2.247	3.631	3.546	2.259	0.000
9	0.635	0.662	0.779	1.061	1.610	2.470	2.271	0.000	0.000

APPENDIX D

Input description

Input description for the steady-state version of NEM: NEM/2 (2 dimensions) and NEM/3 (3 dimensions)

The only difference between the two codes is the way the arrays are dimensioned. In NEM/2 problems with up to 68 × 86 nodes can be handled, and in NEM/3 problems with up to 12 × 12 × 20 nodes can be handled, both with 2-energy groups. The array limits are only examples and they can be changed within the main programs. The absolute limits are

(no. of nodes in x) × (no. nodes in y) × (no. of nodes in z) × (no. of energy groups) ≤ 13107

(This limitation comes from the overall limit in array sizes in Fortran IV at 65535 elements).

Files in use

5	kind = reader:	Input to the program
6	kind = printer:	Output to print
10	kind = disk:	Dump for restart
11	kind = disk:	Plotfile for horizontal plot at x = 0
12	kind = disk:	Plotfile for horizontal plot at x = y
13	kind = disk:	Plotfile for vertical plot

Input from file 5

Variable	Format	Description
TEXT	13A6	The heading text
I	7X,I3	Maximum number of nodes in x-direction
J	7X,I3	Maximum number of nodes in y-direction
K	7X,I3	Number of nodes in z-direction
N	7X,I3	Total number of nodes
G	7X,I3	Number of energy groups I*J*K*G < 13107
NNODE	7X,I3	Number of different fuel types
NAPP	7X,I3	Approximation order for expansion-polynomial (2,3,4)
NSTART	7X,I3	If NSTART = 0 no dump in made If NSTART = 1 a dump is made after TDUMP seconds of processing time If NSTART = 2 restart from dump, no new dump is made If NSTART = 3: restart from dump, a new dump is made after TDUMP seconds.
TDUMP	7X,E10.3	time for dump in seconds of processing time

The following data are read only if NSTART = 0 or 1.

Variable	Format	Description
ISYM	7X,I3	Characterizes symmetry properties of the core If ISYM = 1: full core If ISYM = 2: 1/2 core - mirror symmetry If ISYM = 3: 1/2 core - rotational symmetry If ISYM = 4: 1/4 core - mirror symmetry If ISYM = 5: 1/4 core - rotational symmetry If ISYM = 6: 1/8 core
NSYM	7X,I3	Indicate the lines of symmetry in more detail If NSYM = 0: symmetry axes at node edges If NSYM = 1: I-symmetry axis at node edges J-symmetry axis through node centers If NSYM = 2: I-symmetry axis through node centers J-symmetry axis at node edges If NSYM = 3: symmetry axes through node centers
	80X	
IMIN(1)	14I5	Start nodes in x-direction
-		
IMIN(J)		If ISYM > 2 IMIN(i) = 1 for i = 1,(1)J
	80X	
IMAX(1)	14I5	End nodes in x-direction
-		
IMAX(J)		
	80X	
JMIN(1)	14I5	Start nodes in y-direction
-		
JMIN(I)		
	80X	

Variable	Format	Description
JMAX(1)		End nodes in y-direction
-		
JMAX(I)		
	80X	
NODE(1,JMIN(1),1)		
	14I5	Fuel types for each node
NODE(1,JMIN(1)+1,1)		Only one row in one layer on each card
-		
NODE(1,JMAX(1),1)		The total number of inputs is N
NODE(1,JMIN(2),2)		
	14I5	
-		
NODE(1,JMAX(2),2)		
-		
NODE(1,JMAX(I),I)		
NODE(2,JMIN(1),1)		
	14I5	
-		
NODE(K,JMAX(I),I)		
DX	7X,E10.3	Node-dimension in x-direction in cm
DY	7X,E10.3	Node-dimension in y-direction in cm
DZ	7X,E10.3	Node-dimension in z-direction in cm

The following inputs are repeated until data for all fuel types are read.

Variable	Format	Description
NUMBER	7X,I3	Fuel type number
D(1,NUMBER)		
	7E10.3	Diffusion constants in cm for
-		fuel type no NUMBER
D(G,NUMBER)		
SIGA(1,NUMBER)	7E10.3	Absorption cross sections in cm^{-1}
-		
SIGA(G,NUMBER)		
SIGG(1,1,NUMBER)	7E10.3	Scattering cross sections in cm^{-1}
SIGG(1,2,NUMBER)		SIGG(G1,G2,NUMBER) is scatttering cross
		section from group G2 to group G1
-		in fuel type no NUMBER
SIGG(1,G,NUMBER)		
SIGG(2,1,NUMBER)		
-		
SIGG(2,G,NUMBER)		
-		
SIGG(G,G,NUMBER)		
USIG(1,NUMBER)		
	7E10.3	ν * fission cross sections in cm^{-1}
-		
USIG(G,NUMBER)		

Variable	Format	Description
NY(1)	7E10.3	ν
-		
NY(G)		
IBUCK	7X,I3	If IBUCK = 0 the albedos in the z-direction are used If IBUCK = 1 the buckling is used instead of the albedos in the z-direction
	80X	
ALBXL(1,1,1)	7E10.3	Albedo-matrices for the x-direction at the left edge.
-		
ALBXL(1,G,1)		Different albedo-matrices can be used for
ALBXL(2,1,1)		different "channels".
-		
ALBXL(G,G,1)		Only albedo-matrices for
-		one channel at each card
ALBXL(1,1,2)	7E10.3	
ALBXL(G,G,J)	7E10.3	If ISYM > 2 ALBXL is not used
	80X	
ALBXR(1,1,1)	7E10.3	Albedo-matrices for the x-direction at right edge. If ISYM = 6 some of the albedos are not used.
-		
ALBXR(G,G,J)		
	80X	
ALBYL(1,1,1)	7E10.3	Albedo-matrices for the y-direction at left edge. If ISYM > 3 ALBYL is not used
-		
ALBYL(G,G,I)		
	80X	
ALBYR(1,1,1)	7E10.3	Albedo-matrices for the y-direction at right edge.
-		
ALBYR(G,G,I)		

If BUCK = 0 the following cards are read

Variable	Format	Description
	80X	
ALBZL(1,1) - ALBZL(1,G) - ALBZL(G,G)	7E10.3	Albedo-matrix for the z-direction at left edge (lower end)
	80X	
ALBZR(1,1) - ALBZR(G,G)	7E10.3	Albedo-matrix for the z-direction at right edge (upper end, top)

If IBUCK = 1 the following cards are read

Variable	Format	Description
	80X	
BUCK - BUCK(G)	7E10.3	The energy group-dependent bucklings

Variable	Format	Description
NPART	7X,I3	Number of refinements
MAXIN	7X,I3	Maximum number of inner iterations per outer
MAXOUT	7X,I3	Maximum number of outer iterations
LEAK	7X,I3	Order of the transverse leakage approximation
UPDATE	7X,I3	Number of inner iterations between calculating the coefficients. Usual 3.
MODE	7X,I3	If MODE = 0 no Chebyshev-acceleration (ω=0) and the multiplication factor is calculated by the sum of fluxes in group 1: If MODE = 1 Chebyshev-acceleration in inner iterations ($\omega=1-\gamma\cdot\cos(\pi/2\cdot IM/MAXIN)$ where IM = { ITNO ITNO odd; 2*MAXIN+1-ITNO ITNO even and the multiplication factor is calculated by the sum of fluxes in group 1. If MODE = 2 no Chebyshev-acceleration, and the multiplication factor is calculated by the maximum flux in group 1. If MODE = 3 Chebyshev-acceleration, and the multiplication factor is calculated by the maximum flux in group 1
GAMMA	7X,E10.3	The acceleration constant in the formula for Chebyshev acceleration. 0.7-0.8 is recommended
EIGV	7X,E10.3	Start value of the eigenvalue (normally 1.0)
DL	7X,E10.3	Convergence criteria for convergence in the eigenvalue ~ 10^{-5}
DF	7X,E10.3	Convergence criterion for the flux

In the following 13 output options the values are

If OPTION = 0 no output

If OPTION = 1,2,..,G output of group: OPTION

If OPTION = 9 output of all groups

Variable	Format	Description
OPTION(1)	7X,I1	Output of flux
OPTION(2)	7X,I1	Output of incoming currents in x-direction at left edge
OPTION(3)	7X,I1	Output of incoming currents in x-direction at right edge
OPTION(4)	7X,I1	Output of incoming currents in y-direction at left edge
OPTION(5)	7X,I1	Output of incoming currents in y-direction at right edge
OPTION(6)	7X,I1	Output of incoming currents in z-direction at left edge
OPTION(7)	7X,I1	Output of incoming currents in z-direction at right edge
OPTION(8)	7X,I1	Output of outgoing currents in x-direction at left edge
OPTION(9)	7X,I1	Output of outgoing currents in x-direction at right edge
OPTION(10)	7X,I1	Output of outgoing currents in y-direction at left edge
OPTION(11)	7X,I1	Output of outgoing currents in y-direction at right edge
OPTION(12)	7X,I1	Output of outgoing currents in z-direction at left edge
OPTION(13)	7X,I1	Output of outgoing currents in z-direction at right edge

Variable	Factor	Description
OPTION(14)	7X,I1	If OPTION(14)=1 output of fuel types
OPTION(15)	7X,I1	If OPTION(15) = 1 output of eigenvalue for each outer iteration
OPTION(16)	7X,I1	If OPTION(16) = 1 output of multiplication factor for each inner iteration
OPTION(17)	7X,I1	If OPTION(17) = 1 output of start values (those defined in option 1 to 14 and 21 to 61)
OPTION(18)	7X,I1	If OPTION(18) > 0 number of outer iteration between output of option 1-14, 21-60
OPTION(19)	7X,I1	If OPTION(19) = 1 output after each inner iteration
OPTION(20)	7X,I1	If OPTION(20) = 1 output at end
OPTION(21)	7X,I1	If OPTION(21) = 1 output of power
OUTPUT(22) -	7X,I1	If OPTION(22) = 1 output of layer 1 (option 1-14 and 21)
OPTION(K+21)	7X,I1	If OPTION(K+21) = 1 output of layer K (option 1-14 and 21)
OPTION(62)	7X,I1	If OPTION(62) = 1 output to plotfile of horizontal view
OPTION(63)	7X,I1	If OPTION(63) = 1 output to plotfile of horizontal view at x = 0. The output variable are those defined in option 1-14, 21-41
OPTION(64)	7X,I1	If OPTION(64) = 1 output to plotfile of horizontal view at x = y
OPTION(65)	7X,I1	Number of channels in which a vertical view is written to plotfile 13. The output variables are those defined in option 1-14, 21
OPTION(66)	7X,I3	Channel number for output to plotfile. If the nodes in the bottom is (1,j,i)
OPTION(OPTION(65)+65)	7X,I3	the channel number is $\sum_{n=1}^{i-1}[JMAX(n)-JMIN(n)+1]+j$

Variable	Format	Description
OPTION(80)	7X,I3	If OPTION(80) = 0 no reference solution If OPTION(80) = 1 a reference solution is read and can be compared with the calculated one if power output is defined (OPTION(21))

The following cards are only read if option(80) = 1

EO	7X,F10.6	The eigenvalue of the reference solution
	80X	
REF(JMIN(1),1)	710.3	The normalizewd power distribution of
REF(JMIN(1)+1,1)		the reference solution. The normalization
-		is only over powers > 0. For
REF(JMAX(1),1)		three-dimensional problems it is the
REF(JMIN(2),2)		assembly power distribution.
-		Only one row on each card
REF(JMAX(2),2)		
-		
REF(JMAX(I),I)		

The following cards are read if NPART > 0 and they are repeated NPART times

NFACTX	7X,I3	Refinement factor in the x-direction
NFACTY	7X,I3	Refinement factor in the y-direction
NFACTZ	7X,I3	Refinement factor in the z-direction

Input description for the dynamic part of NEM

First, the kind of transient has to be incorporated in the program in form of a subroutine PARAM which gives the cross section variation as a function of time

The subroutine has to be declared:

```
SUBRUTINE   PARAM(I,J,K,G,NNODE,NODE,D,SIGA,USIG,SIGG)
INTEGER     G,I,J,K,NNODE,NODE(K,J,I)
REAL        D(G,NNODE),SIGA(G,NNODE),USIG(G,NNODE),SIGG(G,G,
            NNODE),COM(8)
COMMON      /DYNCOM/COM
```

The parameters are

G	Number of energy groups
I	Maximum number of nodes in x-direction
J	Maximum number of nodes in y-direction
K	Maximum number of nodes in z-direction
NNODE	Number of different fuel types
NODE	Fuel type in each node
D	Diffusion constants for each group and fuel type
SIGA	Absorption cross section for each group and fuel type
USIG	$\nu * \Sigma_f$ for each group and fuel type
SIGG	Scattering cross section for each fuel type
COM(1)	Time step just now
COM(2)	Current time
COM(3)-COM(8)	Not used

The subroutine can be bound on the main program

First, a steady-state run has to be made with NSTART = 1 corresponding to a dump for restart.

The files in use are the same as for the steady-state program.

Input from file 5.

Variable	Format	Description
TEXT	13A6	The heading text
NSTART	7X,I3	NSTART = 2, restart from dump
L	7X,I.	Number of precursor families 1 ≤ L ≤ 6
	80X	
	80X	
V(1)	7E10.3	The neutron velocities in each group
-		
V(G)		
	80X	
BETA(1)	7E10.3	The β-value for each precursor family
-		
BETA(L)		
	80X	
DECAY(1)	7E10.3	The decay constants for each
-		precursor family
DECAY(L)		
DT	7X,E10.3	Guess on the first time step
DTMIN	7X,E10.3	Minimum allowed time step
TEND	7X,E10.3	The end time for the integration
MAXIN	5X,I5	The maximum number of inner iterations, before halving the time step
ITGOOD	7X,I3	Minimum number of time steps before time step increase is attempted.
MODE	7X,I3	If MODE = 2 the convergence criterion for convergence in inner iterations has to be kept for the fluxes in all nodes and energy groups. If MODE = 2 the convergence criterion is used on a weighted sum of fluxes
DF	7X,E10.3	Convergence criterion for the flux in the inner iterations
POWER	7X,E10.3	The nominal reactor power

In the following 13 output options the values are

if OPTION = 0	no output
if OPTION = 1,2, ..,G	output of group: OPTION
if OPTION = 9	output of all groups

Variable	Format	Description
OPTION(1)	7X,I1	Output of flux
OPTION(2)	7X,I1	Output of incoming currents in r-direction at left edge
OPTION(3)	7X,I1	Output of incoming currents in x-direction at right edge
OPTION(4)	7X,I1	Output of incoming currents in y-direction at left edge
OPTION(5)	7X,I1	Output of incoming currents in y-direction at right edge
OPTION(6)	7X,I1	Output of incoming currents in z-direction at left edge
OPTION(7)	7X,I1	Output of incoming currents in z-direction at right edge
OPTION(8)	7X,I1	Output of outgoing currents in x-direction at left edge
OPTION(9)	7X,I1	Output of outgoing currents in x-direction at right edge
OPTION(10)	7X,I1	Output of outgoing currents in y-direction at left edge
OPTION(11)	7X,I1	Output of outgoing currents in y-direction at right edge
OPTION(12)	7X,I1	Output of outgoing currents in z-direction at left edge
OPTION(13)	7X,I1	Output of outgoing currents in z-direction at right edge

Variable	Format	Description
OPTION(14)	7X,I1	If OPTION(14)=1 output of fuel types
OPTION(15)	7X,I1	Not used
OPTION(16)	7X,I1	If OPTION(16) = 1 output of step length nominal reactor power etc. for each time step
OPTION(17)	7X,I1	If OPTION(17) = 1 output of start values (those defined in option 1 to 14 and 21 to 61)
OPTION(18)	6X,I5	If OPTION > 0: Option(18)*DT time between output of distributions (chosen by OPTION(1) - OPTION(14) and OPTION(21)-OPTION(61)). If OPTION(18) = 0 no output
OPTION(19)	7X,I1	If OPTION(19) = 1 output after each inner iteration
OPTION(20)	7X,I1	If OPTION(20) = 1 output at end
OPTION(21)	7X,I1	If OPTION(21) = 1 output of power
OPTION(22)	7X,I1	If OPTION(22) = 1 output of layer 1 (OPTION 1-14 and 21)
-		
OPTION(K+21)	7X,I1	If OPTION(K+21) output of layer K (OPTION 1-14 and 21)
OPTION(62)	7X,I1	If OPTION(62) = 1 output to plotfile of horizontal view
OPTION(63)	7X,I1	If OPTION(63) = 1 output to plotfile 11 of horizontal view at x = 0. The output-variables are those defined in OPTION 1-14 and 21-41
OPTION(64)	7X,I1	If OPTION(64) = 1 output to plotfile 12 of horizontal view at x = y
OPTION(65)	7X,I1	Number of channels in which a vertical view is written to plotfile 13. The output variables are those defined in OPTION 1-1 and 21.

Variable	Format	Description
OPTION(66)	7X,I3	Channel number for output to plotfile.
-		If the nodes in the botton is (1,j,i)
OPTION(OPTION(65)+65)		the channel number is
	7X,I3	$\sum_{n=1}^{i-1} (JMAX(n) - JMIN(n)+1) + j$
OPTION(80)	7X,I3	Not used.

APPENDIX E

E.1 Input example for steady-state calculation.

```
?BEGIN JOB LMWLWR;CLASS=1;CHARGE=3020501
?MAXPROCTIME=300;MAXIOTIME=300;MAXLINES=2000
?DESTNAME=RTA
?RUN OBJECT/NEM/LMWLWR3D/A
?FILE FILE5(KIND=READER)
?FILE FILE6(KIND=DISK,MAXRECSIZE=22,BLOCKSIZE=330)
?FILE FILE6(*,TITLE=LMWLWR/OUTPUT,PROTECTION=SAVE)
?FILE FILE10(TITLE=DUMP/LMWLWR3D)
?FILE FILE11(TITLE=PLOT/HORIZONTAL/IO/LMWLWR3D)
?FILE FILE12(TITLE=PLOT/HORIZONTAL/IJ/LMWLWR3D)
?FILE FILE13(TITLE=PLOT/VERTICAL/LMWLWR3D)
?DATA
```

```
LMW LWR TRANSIENT BENCHMARK. 1/8 CORE. 200 NODES. 20 CM NODES.
NSTART    1
IMAX      5
JMAX      6
KMAX     10
N       200
GROUPS    2
NNODE     4
NAPP      4
TDUMP   2.600E+02
ISYM      6
NSYM      3
IMIN
    1    1    1    1    1    1
IMAX
    1    2    3    4    5    5
JMIN
    1    2    3    4    5
JMAX
    6    6    6    6    6
NODE
    4    4    4    4    4    4
    4    4    4    4    4
    4    4    4    4
    4    4    4
    4    4
    1    1    1    1    3    4
    1    1    1    3    4
    1    1    3    4
    3    3    4
    4    4
    1    1    1    1    3    4
    1    1    1    3    4
    1    1    3    4
    3    3    4
    4    4
```

```
1    1    1    1    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
1    1    1    1    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
1    1    1    2    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
1    1    1    2    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
1    1    1    2    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
1    1    1    2    3    4
1    1    1    3    4
1    1    3    4
3    3    4
4    4
2    4    4    2    4    4
4    4    4    4    4
2    4    4    4
4    4    4
4    4
```

```
DX       2.000E+01
DY       2.000E+01
DZ       2.000E+01
FUEL TYPES: D, SIGA, SIGG, USIG
TYPE NR  1
  1.423913  0.356306
0.010402060.08766207
 0.000E+00 0.000E+00 0.0175555 0.000E+00
.006477691 0.1127328
TYPE NR  2
  1.423913  0.356306
0.010952060.09146217
 0.000E+00 0.000E+00 0.0175555 0.000E+00
0.00647769 0.1127328
TYPE NR  3
  1.425611  0.350574
0.010992630.09925634
 0.000E+00 0.000E+000.01717768 0.000E+00
.007503284 0.1378004
TYPE NR  4
  1.634227  0.264002
.0026605730.04936351
 0.000E+00 0.000E+000.02759693 0.000E+00
 0.000E+00 0.000E+00
KAPPA
 1.000E+00 0.000E+00
NY       2.500000
IBUCK    0
ALBXL
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
```

```
ALBXR
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
ALBYL
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
ALBYR
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
ALBZL
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
ALBZR
-1.000E+00 0.000E+00 0.000E+00-1.000E+00
NPART     0
MAXIN    17
MAXOUT  100
LEAK       2
UPDATE     3
MODE       0
GAMMA    0.000E+00
EIGV     1.000E+00
DL       1.000E-06
DF       1.000E-04
FI       0
CINXL    0
CINXR    0
CINYL    0
CINYR    0
CINZL    0
```

```
CINZR  0
CXL    0
CXR    0
CYL    0
CYR    0
CZL    0
CZR    0
NODE   0
EIGV   1
FACT   0
START  0
OUTER  0
INNER  0
END    1
POWER  1
K1PLOT 0
K2PLOT 0
K3PLOT 0
K4PLOT 0
K5PLOT 0
K6PLOT 0
K7PLOT 0
K8PLOT 0
K9PLOT 0
K10PLOT0
RADPLOT0
IOPLOT 0
IJPLOT 0
ANTKAN 0
REF    0
?END JOB
```

E.2 Input example for dynamic calculation.

```
?BEGIN JOB LMWLWR/DYN;CLASS=4;CHARGE=3020501
?MAXPROCTIME=4000;MAXIOTIME=60;MAXLINES=5000
?DESTNAME=RTA
?RUN OBJECT/NEM/LMWLWR3D/A
?FILE FILE5(KIND=READER)
?FILE FILE6(KIND=DISK,MAXRECSIZE=22,BLOCKSIZE=330)
?FILE FILE6(*,TITLE=LMWLWR/OUTPUT/DYN,PROTECTION=SAVE)
?FILE FILE7(TITLE=FILE7/LMWLWR3D,PROTECTION=SAVE,NEWFILE=TRUE)
?FILE FILE10(TITLE=DUMP/LMWLWR3D)
?FILE FILE11(TITLE=PLOT/HORIZONTAL/IO/LMWLWR3D)
?FILE FILE12(TITLE=PLOT/HORIZONTAL/IJ/LMWLWR3D)
?FILE FILE13(TITLE=PLOT/VERTICAL/LMWLWR3D)
?DATA
```

```
LMWLWR 3D REACTOR KINETICS BENCHMARK. DYNAMIC.
NSTART   2
L        6
DYNAMIC INPUT
NEUTRON VELOCITIES
 1.250E+07 2.500E+05
BETAS
 2.470E-04 0.0013845 1.222E-03 0.0026455 8.320E-04 1.690E-04
DECAY CONSTANTS
 1.270E-02 3.170E-02 1.150E-01 3.110E-01 1.400E+00 3.870E+00
DT       2.500E-01
DTMIN    1.000E-03
TEND     6.000E+01
MAXIN    20
ITDOUB   10
MODE      2
DF       1.000E-02
POWER    1.500E+02
FI      0
CINXL   0
CINXR   0
CINYL   0
CINYR   0
CINZL   0
CINZR   0
CXL     0
CXR     0
CYL     0
CYR     0
CZL     0
CZR     0
NODE    0
EIGV    1
FACT    1
START   1
OUTER     40
INNER   0
END     1
POWER   1
```

```
K1PLOT 0
K2PLOT 0
K3PLOT 0
K4PLOT 0
K5PLOT 0
K6PLOT 0
K7PLOT 0
K8PLOT 0
K9PLOT 0
K10PLOT0
RADPLOT0
IOPLOT 0
IJPLOT 0
ANTKAN 0
REF    0
?END JOB
```

APPENDIX F

Output example from steady-state calculation

```
LMW LWR TRANSIENT BENCHMARK. 1/8 CORE. 200 NODES. 20 CM NODES.

OUTPUT OF ALL INPUT VARIABLES.


NUMBER OF NODES IN X-DIRECTION =   5
NUMBER OF NODES IN Y-DIRECTION =   6
NUMBER OF NODES IN Z-DIRECTION =  10
NUMBER OF NODES =                200
NUMBER OF ENERGY GROUPS =          2

NODE DIMENSIONS IN CM:

DX =  2.000E+01
DY =  2.000E+01
DZ =  2.000E+01

ISYM = 6   1/8 CORE.
NSYM = 3   SYMMETRY AXES THROUGH NODE CENTRES.

NUMBER OF FUEL TYPES =             4


PICTURE OF FUEL TYPES IN LAYER NO.  1

  X\Y      1     2     3     4     5     6

  1        4     4     4     4     4     4
  2              4     4     4     4     4
  3                    4     4     4     4
  4                          4     4     4
  5                                4     4
```

PICTURE OF FUEL TYPES IN LAYER NO. 2

X\Y	1	2	3	4	5	6
1	1	1	1	1	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 3

X\Y	1	2	3	4	5	6
1	1	1	1	1	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 4

X\Y	1	2	3	4	5	6
1	1	1	1	1	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 5

X\Y	1	2	3	4	5	6
1	1	1	1	1	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 6

X\Y	1	2	3	4	5	6
1	1	1	1	2	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 7

X\Y	1	2	3	4	5	6
1	1	1	1	2	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 8

X\Y	1	2	3	4	5	6
1	1	1	1	2	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 9

X\Y	1	2	3	4	5	6
1	1	1	1	2	3	4
2		1	1	1	3	4
3			1	1	3	4
4				3	3	4
5					4	4

PICTURE OF FUEL TYPES IN LAYER NO. 10

X\Y	1	2	3	4	5	6
1	2	4	4	2	4	4
2		4	4	4	4	4
3			2	4	4	4
4				4	4	4
5					4	4

PARAMETERS FOR FUEL TYPE NO. 1

DIFFUSION COEFFICIENTS IN CM:

1.424E+00 3.563E-01

ABSORPTION CROSS SECTIONS IN CM**(-1):

1.040E-02 8.766E-02

SCATTERING CROSS SECTIONS IN CM**(-1):

0. 0. 1.756E-02 0.

NY*FISSION CROSS SECTIONS IN CM**(-1):

6.478E-03 1.127E-01

PARAMETERS FOR FUEL TYPE NO. 2

DIFFUSION COEFFICIENTS IN CM:

1.424E+00 3.563E-01

ABSORPTION CROSS SECTIONS IN CM**(-1):

1.095E-02 9.146E-02

SCATTERING CROSS SECTIONS IN CM**(-1):

0. 0. 1.756E-02 0.

NY*FISSION CROSS SECTIONS IN CM**(-1):

6.478E-03 1.127E-01

PARAMETERS FOR FUEL TYPE NO. 3

DIFFUSION COEFFICIENTS IN CM:

1.426E+00 3.506E-01

ABSORPTION CROSS SECTIONS IN CM**(-1):

1.099E-02 9.926E-02

SCATTERING CROSS SECTIONS IN CM**(-1):

0. 0. 1.718E-02 0.

NY*FISSION CROSS SECTIONS IN CM**(-1):

7.503E-03 1.378E-01

PARAMETERS FOR FUEL TYPE NO. 4

DIFFUSION COEFFICIENTS IN CM:

1.634E+00 2.640E-01

ABSORPTION CROSS SECTIONS IN CM**(-1):

2.661E-03 4.936E-02

SCATTERING CROSS SECTIONS IN CM**(-1):

0. 0. 2.760E-02 0.

NY*FISSION CROSS SECTIONS IN CM**(-1):

0. 0.

KHI:

1.000E+00 0.

NY = 2.500E+00

IBUCK: 0

ALBEDOS:

X-DIRECTION - LEFT EDGE:

-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00

X-DIRECTION - RIGHT EDGE:

-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00

Y-DIRECTION - LEFT EDGE:

-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00

Y-DIRECTION - RIGHT EDGE:

-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00
-1.000E+00	0.	0.	-1.000E+00

Z-DIRECTION - LEFT EDGE:

-1.000E+00	0.	0.	-1.000E+00

Z-DIRECTION - RIGHT EDGE:

-1.000E+00	0.	0.	-1.000E+00

```
TIME LIMIT FOR DUMP =      2.600E+02 SECONDS

APPROXIMATION ORDER =                 0
NUMBER OF REFINEMENTS =               0
NUMBER OF INNER ITERATIONS =         17
NUMBER OF OUTER ITERATIONS =        100
LEAKAGE APPROXIMATION ORDER =         2
ITERATIONS BETWEEN UPDATING =         3
MODE =                                0
GAMMA =                      0.
EIGENVALUE =                 1.000E+00
DEVIATION ON EIGENVALUE =    1.000E-06
DEVIATION ON FLUX =          1.000E-04

OUTPUT OPTIONS :

15 : OUTPUT OF EIGENVALUE FOR EACH OUTER ITERATION
20 : OUTPUT AT END
21 : OUTPUT OF POWER
```

LMW LWR TRANSIENT BENCHMARK. 1/8 CORE. 200 NODES. 20 CM NODES.

1	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9824223
2	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9882730
3	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9934466
4	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9962038
5	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995961
6	OUTER	20	INNER	ITERATIONS	EIGENVALUE 0.9988922
7	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9990594
8	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9992524
9	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9993698
10	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995440
11	OUTER	20	INNER	ITERATIONS	EIGENVALUE 0.9995017
12	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995092
13	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995192
14	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995260
15	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995371
16	OUTER	20	INNER	ITERATIONS	EIGENVALUE 0.9995337
17	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995341
18	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995347
19	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995351
20	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995990
21	OUTER	20	INNER	ITERATIONS	EIGENVALUE 0.9995461
22	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995373
23	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995360
24	OUTER	17	INNER	ITERATIONS	EIGENVALUE 0.9995358

OUTER ITERATION NUMBER = 24 INNER ITERATION NUMBER = 17

PICTURE OF POWER IN HORIZONTAL LAYER.

X\Y	1	2	3	4	5	6
1	1.560	1.476	1.283	0.982	0.642	0.000
2		1.417	1.244	0.964	0.626	0.000
3			1.126	0.873	0.554	0.000
4				0.763	0.380	0.000
5					0.000	0.000

PICTURE OF POWER IN VERTICAL DIRECTION.

10	0.015
9	0.451
8	0.884
7	1.222
6	1.418
5	1.446
4	1.278
3	0.934
2	0.469
1	0.000

AFTER 24 OUTER ITERATIONS THE PROGRAM IS TERMINATED WITH A DUMP ON FIL

Sales distributors:
G.E.C. Gad Strøget
Vimmelskaftet 32
DK-1161 Copenhagen K, Denmark

Available on exchange from:
Risø Library, Risø National Laboratory,
P.O.Box 49, DK-4000 Roskilde, Denmark

ISBN 87-550-1169-1
ISSN 0106-2840